Hálózati alapismeretek

Uhlár László

2011. október 21.

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 iroda házában. 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 (tárolók, nyomtatók) 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.

- 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.

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(IMP, Interface Management Processor) áll, amelyeket adathálózat köt össze és néhány IMP 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. „Rétegek”

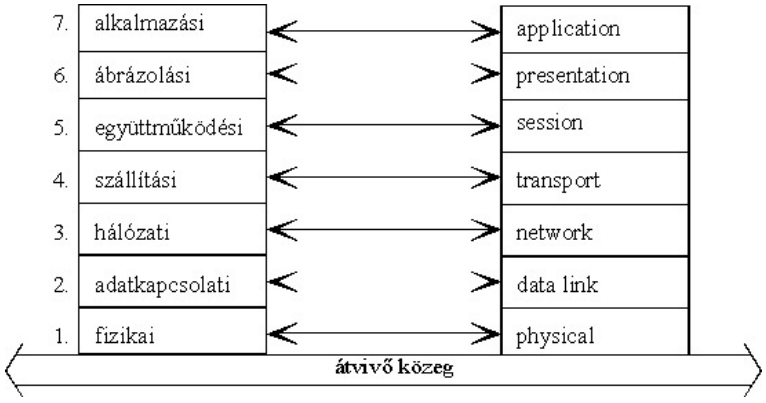
2.1. Mi ez?

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. Egy program kellene, hogy gondoskodjon a megfelelő feszültségszintek előállításától kezdve a célzott gép azonosításáig mindenről. Ha a hálózati kábelezésen változtatnak és a réz vezeték helyett üvegszálat használnának, az egész programot újra kellene írni, ki kellene vserélni. Ezért kialakult az a fajta modellezése a hálózati kommunikációnak, amikor logikailag több részre bontják a folyamatot, 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 levelezésnél 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.

2.2. ISO OSI

A Nemzetközi Szabványügyi Szervezet (ISO, International Standard Organization) létrehozott egy ajánlást (nem szabvány!!!), amelyet OSI (Open System Interconnect - nyílt rendszerek összekapcsolása) modellnek hívnak, és amely hét rétegre bontja logikailag a számítógép hálózatok működését.
Segítheti a megértést, ha két magasrangú államférfira gondolunk, akik tol-



1. ábra. Az OSI rétegek

mácsok segítségével kommunikálnak egymással: az egyik vezető a saját tolmácsának mondja, az elmondja a másik tolmácsnak, az lefordítja a saját főnökének. Kik kommunikálnak egymással: a vezetők beszélni a saját tolmácsaikkal beszélnek, de mégis a két államférfi cserél eszmét a beszélgetés sorá.
A számítógép-hálózatokon folyó kommunikáció során is az egyes egymásnak megfelelő rétegek kommunikálnak logikailag, csak ennek során kihasználják az alattuk lévő réteg szolgáltatásait, amely esetleg szintén kihasználja az alatta lévő rétegét, stb. A tényleges fizikai jeltovábbítás a fizikai rétegen történik.

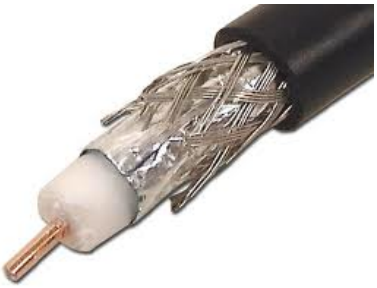
3. Az egyes rétegek feladata

3.1. A fizikai réteg

A tényleges fizikai jeltovábbítás terepe. Sok fontos dolgot kell ennek a rétegnek megoldania. A számítástechnikában általánosan használt digitális jeleket kell valahogyan az adott közegen továbbítani illetve fogadni.
Itt említendő fontos fogalom a baud. Értéke megmutatja a másodpercenkénti jelváltások számát az adott közegen. Ez nem feltétlenül egyezik meg a másodpercenként átküldött bitek számával, hiszen ha egy jelváltás valamilyen alkalmas tömörítő kódolással négy bit információt tud hordozni, akkor egy 1000 baud-os vonal 4000 bitet visz át másodpercenként.
Másik fontos fogalom az átviteli sebesség. Mértékegysége a bit per secundum (bps): 10 bps a sebesség, ha 10 bit információ továbbítódik egy másodperc alatt. Elterjedt prefixált mértékegységek még a kbps: 256 kbps a sebesség, ha 256 ezer bit információ továbbítódik másodpercenként, az Mbps illetve a Gbps hasonlóan megabit illetve gigabit egységekben. Nem összekeverendő a MBps-el, ami megabájtban adja meg az átvitt adatmennyiséget.
Szintén megemlítendő fogalom a szinkronizáció: egy megfelelő jelszint 1 másodpercen keresztül értelmezhető 128000 db egyesnek (128kbps vonal), de értelmezhető 256000 db egyesnek is (256 kbps vonal).
Gyakori, hogy a közeg analóg jelek továbbítására képes (analóg: valamilyen fizikai jellemző nagysága hordozza az információt), ekkor meg kell oldani a digitális jelek (valamely fizikai jellemző léte vagy nem léte hordozza a jelet) analóg jellé konvetálását(DAC: digitális analóg konverter), majd a vonal túlvégén az érkező analóg jeleket „digitalizálni" kell (ADC: analóg digitális konverter).
Az analóg-digitális konverzióval sok helyen találkozhatunk (például hang digitalizálás). Aki utána szeretne jobban olvasni, ajánlható a következő kis leírás: http://studio.pataky.hu/edu/14p/atvitel/PCM.pdf. Érdemes lehet utánajárni a PCM, DSD, PWM rövidítések jelentésének is.

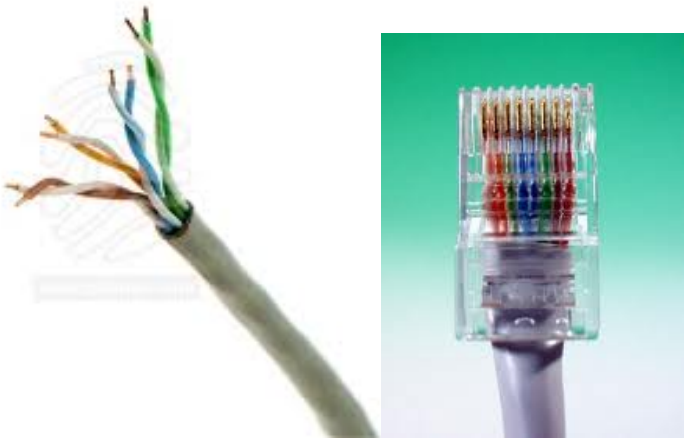
3.1.1. A „vezeték"

–A koaxiális kábel olyan, a híradástechnikában használt vezetéktípus, amely egy belső vezető érből, dielektrikumból, fémhálóból és külső szigetelésből áll. A fémháló szerepe az árnyékolás, azaz a belső éren továbbított jelek megóvása a külső zavaroktól. Elsősorban rádiófrekvenciás jelek továbbítására használják.
–UTP (Unshielded Twisted Pair: árnyékolatlan csavart érpár). Felhaszná-



2. ábra. Koaxiális kábel

lóként ezzel a kábellel találkozunk napjainkban a legtöbbet. 8 szál vezeték párosával egymás köré tekerve alkotja a kábelt, a koaxszal ellentétben nincs külön árnyékolása. Az egymás köré tekeréssel csökkentik a környezet zavaró jeleit. A csatlakozóban a kábelek sorrendje fontos, nem mindegy, hogy mely



(a) UTP kábel (b) UTP csatlakozó

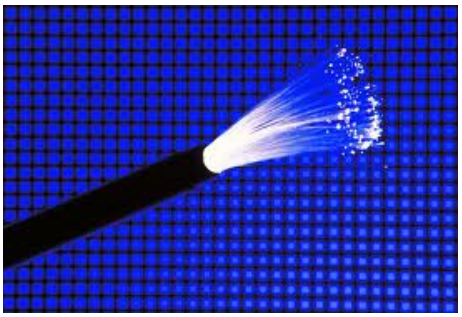
3. ábra. A „csavart érpár”

színű vezeték hová kerül. Készülhet egyenes bekötésű illetve „cross link” kábel is. A mai modern eszközök automatikusan érzékelik, hogy milyen kábelt csatlakoztattunk és ennek megfelelően veszik használatba.
Az UTP kábeleket több kategóriába sorolják a paramétereik szerint. Ezek:

- CAT1 - telefonkábel (hangátvitel, 2 érpár)
- CAT2 - maximum 4 Mb/s adatátviteli sebesség érhető el vele.

- CAT3 - 10 Mb/s az adatátviteli sebessége. Csillag topológiánál alkalmazzák, ethernet hálózatokban (Legacy Ethernet[10MB/s-os] közege).
- CAT4 - max. 20 Mb/s adatátviteli sebességű.
- CAT5 - 100 Mb/s adatátviteli sebességű, csillag topológiánál alkalmazzák, ethernet hálózatokban.
- CAT5e, CAT6 - 1000 Mb/s átviteli sebesség.

–A jelenlegi legkorszerűbb vezetékes adatátviteli módszer az üvegszál vagy más néven optikai technológia alkalmazása. Üvegszálas hálózat kiépítésére akkor kerül sor, ha különösen nagy elektromágneses hatások érik a vezetékeket vagy nagy távolságokat kell áthidalni. Itt a fényáteresztő anyagból készült optikai szálon tovahaladó fényimpulzusok szállítják a jeleket. Az optikai kábel egy olyan vezeték, amelynek közepén üvegszál fut. Ezt az üvegszálat gondosan kiválasztott anyagú burkolat veszi körül. A különleges anyag tulajdonsága, hogy az ide-oda cikázó fény sohasem tudja elhagyni a kábelt. Ezért a fény a vezeték elején lép be és a végén lép ki belőle. De így is meg kell erősíteni és újra kell rendezni a fényt. A legnagyobb áthidalható távolság manapság 80 kilométer, ami lényegesen hosszabb táv a hasonló rendű kábelekhez képest. Az adó, ami lehet LED vagy lézer, elektronikus adatot küld át a kábelen melyet előzőleg fotonná alakítottak. A fotonok hullámhosszai az 1200-1500-ig terjedő nanométer spektrumban lehetnek.
Az optikai átviteli rendszer három komponensből áll: az átviteli közeg-ből (hajszálvékony üveg vagy szilikát), amit egy szilárd fénytörő réteg véd (szintén üveg vagy műanyag), a fényforrásból (LED vagy lézerdióda) és a fényérzékelőből (fotodióda).



4. ábra. Üvegszál

–WIFI: vezeték nélküli mikrohullámú kommunikációt megvalósító szabvány (IEEE802.11). Napjainkban a mobil eszkzök (laptop) elterjedésével egyre

népszerűbb a felhasználók körében. A kábeles kommunikációnál nagyobb hangsúlyt kapnak a terepviszonyok, távolság, lehallgathatóság.

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 adatátviteli hibáktól mentes szolgáltatást nyújt a hálózati réteg számára. Az ellenőrzés érdekében az adó oldal a bemenő adatokat meghatározott hosszúság darabokra – keretekké- tördeli. Az adatkerteket fel kell ismernünk. Ezért a keretek elé és mögé speciális bitmintákat helyezünk el. A bitmintáink az adatkeret belsejében is előfordulhatnak, ezért ellenőrző eljárásokat kell kitalálnunk. Az adatkapcsolati rétegnek kell feldolgozni az ellenállomásról érkező nyugta - kereteket is. Fontos feladata az adatfolyam vezérlés (flow controll), amelyben a vevőhöz igazítja az adás idejét és sebességét. Felépíti és lebontja a kapcsolatokat, a fogadott adatkereteket a megfelelő fejrész levágása után továbbítja a kijelölt folyamatnak. Fizikai cím (MAC) segítségével azonosítja, hogy mely hálózati interfészen (NIC) kell továbbítani az adatot.

3.2.1. Az ethernet

Több technológia is kialakult a fentiek megoldására, környezetünkben legelterjedtebben az ethernettel találkozhatunk (IEEE 802.3 szabvány). A többiekhez hasonlóan az adatküldésre készülő állomás is „hallja” a közegen zajló forgalomat (Multiple Access). Amennyiben azt érzékeli, hogy már valaki adás állapotban van, azaz vivőt érzékel (Carrier Sense), akkor várakozik az adás megszűnésére. Amikor a közegen később „csend” lesz, akkor megkezdi az adást, amelyet azok az állomások vesznek, amelyek az üzenetben szereplő (MAC) cím alapján megszólítva érzik magukat. Előfordulhat olyan helyzet, hogy (megközelítően) egyszerre többen kezdenek adásba, ekkor az adók egymást zavarják, ezért a közegre került információ sérül. A hálózat ilyen állapotát (ütközést) detektálni kell (Collision Detection). Ezt az állapotot az adók az adott jel visszaolvasásakor érzékelt hibákról ismerik fel („nem hallja tisztán a saját hangját”). Ekkor minden, az ütközést érzékelő adó beszünteti az adást, majd véletlenszerű késleltetési idő után újból próbálkozik az adással. A véletlen időzítést követően valószínű, hogy az egyik adó kellően korán lefoglalja a közeget és a többiek az adásuk megkezdése előtt érzékelik azt.
–Kezdetben egy sínszerűen kialakított koaxiális kábellel megvalósított vezetékre voltak felfűzve a hálózat gépei. Mindenki mindenki adását hallotta, sok lehetett így az ütközés. A technológia miatt rendkívül érzékeny volt a

kábel szakadásokra: bárholt megszakadt a vezeték, az egész sínen megállt a forgalom. A sok ütközés elkerülésére jelenthetett megoldást, ha bridge vagy router szerű eszközzel (lásd később) két vagy több „ütközési tartományra" osztották. Egy-egy tartományon belül a szakadás továbbra is teljes leállást okozott.
–Ennek megoldására találták ki a csillag szerűen egy központi „elosztóba" csatlakozó kábelekkel megvalósított úgynevezett csillag topológiát. Ez a központi eszköz régebben hub volt, ma már inkább switch. Kábelszakadás esetén csak az az egy gép esett ki a kommunikációból. A hub a fizikai rétegben „dolgozik", az egyik csatlakozóján beérkező jelet az összes csatlakozón kiküldi felerősítve (repeater). Ezzel a megoldással az ütközések száma nem csökkent, továbbra is egy ütközési tarttományt alkotnak a gépek. A switch erre is megoldást kínál. A sweitch (kapcsoló) a keretekben található fizikai cím (MAC) alapján dönti el, hogy mely csatlakozón küldi tovább az adatot. A kapcsoló 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.

3.3. A hálózati réteg

A hálózati réteg feladata a csomagok eljuttatása a forrástól a célig. A célig egy csomag valószínûleg több csomópontot is érint. Ehhez természetesen ismerni kell az átviteli hálózat felépítését, azaz a topológiáját, és ki kell választania a valamilyen szempontból optimális útvonalat. Ha a forrás és a cél eltérõ típusú hálózatokban vannak, a réteg feladata a hálózatok közti különbségbõl adódó problémák megoldása. A hálózati réteg tervezésénél az egyik legnagyobb probléma az összeköttetésre módjának meghatározása. Összeköttetés alapú hálózatban forrás és a cél között felépült állandó úton vándorolnak a csomagok gondoljunk a vezetékes telefonálásra, mint analógiára). Összeköttetés mentes hálózatban elvileg minden egyes csomag különbözõ útvonalakat követhet, mivel a csomagok útválasztása egymástól független. Ilyenkor

a csomagoknak tartalmazniuk kell mind a forrás, mind a cél teljes címét.

3.3.1. IP 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, így szoktuk látni ezeket (pontozott négyes jelölés). Ilyen például: 193.224.69.67. A számítógépeinket fizikai vagy logikai szempontok szerint alhálózatba sorolhatjuk. Az egy alhálózatban lévő gépek egymással közvetlenül (útválasztók nélkül) tudnak kommunikálni. Hogy két gép egy alhálózatban található-e, az az IP címéből kell, hogy kiderüljön. Kezdetben a lehetséges címeket osztályokba sorolták, így beszélhetünk A, B, C, stb. osztályú címekről.
–A osztály: az első számjegy (a 32 -ből) 0, utána lévő 7 bit azonosítja a hálózatot, a maradék 24 bit a hálózaton belül az egyes hosztokat. A legkisebb ilyen hálózat azonosító az 1 lehet, a legnagyobb a 127. (a csupa nullát tiltjuk). Egy-egy ilyen hálózatban rengeteg különböző számítógép lehet: kb.: 2^24.
–B osztály: a cím 10-val kezdődik, 16 bit azonosítja a hálózatot, azon belül a maradék 16 bit a számítógépet. Így a legkisebb hálózatszám a 128.0 lehet, a legnagyobb a 191.255.
–C osztály:a cím 110-val kezdődik, 24 bit azonosítja a hálózatot, a maradék 8 a hostokat. A legkisebb ilyen hálózatcím a 192.0.0, a legnagyobb a 223.255.255.
A többi cím nincs kiosztva, speciális célokra fenntartottak.
Minden osztályban kijelöltek olyan címtartományt, amelyek a nyílt interneten nem használhatók (nem „publikus”). Ezeket a privát címeket egy-egy belső, az internettől elzárt alhálózatban lehet használni. Azaz ilyen privát IP címmel lehet, hogy egy időben több számítógép is rendelkezik. Ezek:
–A osztályban: 10.0.0.0, illetve a 127.0.0.0 a visszacsatoló interfész számára fenntartva.
–B osztályban: 172.16.0.0-172.31.0.0.
–C osztályban: 192.168.1.0-192.168.255.0.
Tehát ezeket a címeket csak belső hálózatokban (intranetekben) lehet használni.
Kezdetben az egyes hálózati eszközök tehát az IP címből meg tudták állapítani, hogy két gép egy alhálózaton van-e.(alapértelmezett hálózatszám). Viszont már elég korán felismerték annak a veszélyét, hogy így nem túl gazdaságos a címek kiosztása, pl.: ha valaki megszerzett egy A osztályú címosztályt, akkor 2^24 darab különböző gépet helyezhetne el benne, de ennyire

valószínűleg nincs szükség. Felmerült az igény arra, hogy ezek helyett az alapértelmezett hálózatszámok helyett szabadabban lehessen gazdálkodni a még meglevő címekkel. továbbra is beszélhetünk A, B, stb. osztályú címekről, de már nem magától értetődő, hogy hány bit azonosítja az alhálózatot.

3.3.2. Az alhálózati maszk

Az IP cím mellé megadunk egy újabb 32 jegyű (bites) számot, amely az elején csupa 1-t tartalmaz, utána csupa 0-t. Ezt szintén pontozott négyes jelöléssel. Ilyen például: 255.255.192.0. Sokkal kényelmesebb leírást, használhatóságot biztosít az úgynevezett CIDR (Classless Interdomain Routing) jelölés, amely esetén a pontozott négyes jelölés helyett az elhálózati maszk egyeseinek a számát adják meg. Az előző példával élve 18 jegy azonosítja a hálózatot. (példa később)
Érdemes lehet kicsit eljátszani az alábbi oldalon: http://www.fport.hu/index.php?site=cidr

3.3.3. Példák

1. példa
Legyen egy gép (A gép) IP címe 193.224.69.67, alhálózati maszkja: 255.255.255.192, másként: 193.224.69.67/26. Binárisan:

11000001111000000100010101000011	IP cím
11111111111111111111111111000000	alhálózati maszk

Szeretne kommunikálni a 193.224.69.121 című géppel (B gép). Ehhez először el kell döntenie, hogy egy alhálózatban vannak-e? A fenti táblázatban lévő IP címnek tehát veszi a hálózat azonosító bitjeit (első 26 jegy), azaz képezi a két számmal a logikai ÉS (AND) műveletet. Így az A gép megkapja, hogy az ő hálózat száma:
11000001111000000100010101
A címzett gép (B) címéből szintén veszi az első 26 jegyet, megnézi, hogy egyeznek-e? Igen, tehát egy alhálózatban vannak, közvetlenül kommunikálhat vele.
Ha ethernet hálózatról van szó, akkor az adatok tényleges küldéséhez szükség van a címzett kártyájának MAC (Media Access Control)címére. (Ez az egész világon elvileg egyedi szám, 12 jegyű hexadecimális szám formájában szokták megadni). Hogy egy IP címhez milyen MAC cím tartozik, ennek kiderítése az ARP feladata (Address Resolution Protokol)
Az ARP egy belső táblázatának (ARP cache) adatait használja az IP címhez tartozó MAC cím meghatározásakor. Ha a kérdéses MAC cím nem szerepel az ARP táblában, akkor egy olyan üzenetet küld ki a hálózatra, amelyet minden hálókártya elfogad, mint neki szóló üzenet (broadcast). Az üzenet

tartalmazza a címzett IP címet. Amely gép felismeri a saját IP címét, az egy célzott, már csak a feladónak szóló üzenetben válaszol, elküldi, hogy mi a MAC címe. Ezt a feladó elraktározza az ARP táblában, majd elkezdődhet a kommunikáció. Fordított működést valósít meg a RARP (Reverse ARP): MAC cím alapján próbálja kideríteni az IP címet. Az ARP cache tartalmát lekérdezhetjük az arp paranccsal (root jog!!!). Nézzük meg a man oldalát!

2. 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 alhálóban vannak? Ehhez a címzett címének veszi az első 26 jegyét: 11000001111000000100010100. Látható, hogy különböznek, azaz nem egy alhálóban vannak. Ekkor egy routernek (útvonal választónak) kell továbbítania az adatokat, annak a dolga valahogyan azokat a címzetthez eljuttatni. Minden hálózatba kötött gépnek van routing táblája, amely arról tartalmaz információkat, hogy bizonyos típusú IP címek esetén mely hálózati csatolón (hálókártyán) küldje ki az adatokat. Ennek tartalmaznia kell egy default bejegyzést: amely címről nincs bejegyzés, hogy merre küldjük, azt küldjük erre tovább. Ez a routing tábla lekérdezhető a route utasítással (root jog!!!). Nézzük meg a man oldalát!
Innét már megegyezik a folyamat a fenti példával: kell az alapértelmezett átjáró (default router) MAC címe. Ha benne van az ARP cache-ben, akkor...., különben....

3.3.4. Mindez linuxon

Az egyes interfészeket beállítani az ifconfig utasítással lehet (root jog!!!) Ha minden kapcsoló és paraméter nélkül adjuk ki az ifconfig utasítást, akkor a lentihez hasonló, az adott interfészről adatokat közlő listát kaphatunk.

```
wlan0     Link encap:Ethernet  HWaddr 90:4c:e5:c9:5d:30
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::924c:e5ff:fec9:5d30/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:460101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:310566 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:591973817 (564.5 MiB)  TX bytes:41034374 (39.1 MiB)
```

Gondosan tanulmányozzuk a manualt!
Sajnos az egyes beállítások megváltoztatásához root jog kell, ezért ezeket kipróbálni nem tudjuk. A gép bekapcsolásakor betöltődő beállításokat GNU/Debian

alatt a /etc/network/interfaces fájl tartalmazza. Listázzuk ki a tartalmát! (cat, manual!!!)
Fontos információkhoz juthatunk a traceroute parancs segítségével. Írjuk be: `traceroute 193.224.69.67` . Az adott géphez vezető „úton" útbaeső csomópontok néhány adatát láthatjuk. (manual!!!)

3.3.5. A DHCP

A fentiekből látható, hogy elég bonyolult egy hálózatban az IP címek karbantartása. ki milyet, milyen maszkkal, ne legyen ütközés, stb. Ennek a problémának a megoldására dolgozták ki a DHCP-t. (Dynamic Host Configuration Protokol). Többek között a fenti (IP cím) beállítások automatikus megoldását teszi lehetővé, ezzel segítve a rendszergazda, a felhasználó munkáját.

3.3.6. A DNS

Igen ám, de ha mi beírunk a böngészőbe egy címet, az a legritkább esetben pontozott négyes jelölés, legtöbbször értelmes emberi kifejezések. Most akkor a gép számokkal azonosítja a másikat vagy „emberi" nevekkel? Számokkal!
Hogyan kapja meg a hálózati réteg a cím alapján az IP címet? Erre szolgál a DNS.
Egy világméretű elosztott (nem egy gépen tárolt, hanem részenként szétosztott) adatbázisban találhatóak a névhez IP címet rendelő információk. Ha a gépünknek szüksége van a turdus.itk.ppke.hu gép címére, egy kérdést küld a .hu legfelső szintű (top level) domain valamely DNS szerveréhez (több gépen vannak ugyan azok 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.)
Próbáljuk ki! Egy böngésző címsorába írjuk be: 217.20.138.26! Ezek talán nehezebben megjegyezhetőek, mint az „emberi" elnevezések.

3.3.7. Az IPv6

Napjainkra égető problémává vált, hogy a fentebb vázolt 32 bites IP címzés nem elegendő, elfogynak a kioszthatóa címek. Erre kínál megoldást az IPv6, amely esetén 128 bites címe van minden hálózati csatolónak.

3.3.8. A NAT

Az IPv4-es címek elégtelen volta miatt (nem jut minden gépnek egyedi) dolgozták ki a Hálózati Címfordítást (Network Address Translation), amely esetén az alhálózatunkon használhatunk privát címeket (melyeknek csak az alhálózatunkon kell egyedinek lennie!). Ha a világhálóra akarunk csatlakozni, akkor egy „átjáró" gépen keresztül jutnak ki a csomagjaink, amely gépen futó NAT szolgáltatás privát címeinket kicseréli az ő egyetlen publikus IP címére és így küldi tovább azokat, közben egy táblázatban feljegyzi, hogy mely privát IP-t írta át. Ha jön a válasz, ő kapja, hiszen ez a gép szerepel feladóként, a táblázata alapján továbbküldi a választ a belső hálóba a ténylegesen kommunikáló privát című gépnek.

3.4. A szállítási réteg

A szállítási réteg az OSI modell legbonyolultabb rétege. Fő célja, hogy megbízható, gazdaságos szolgálatot nyújtson a felette lévő rétegeknek. A szállítási réteg transzparens a felső rétegek számára.
A szállítási réteg tipikus feladatai:

- forgalom szabályozás
- multiplexelés
- virtuális áramkörök felügyelete
- hibajavítás
- csomagképzés és csomagok visszaállítása a felsőbb rétegek számára.

A forgalom szabályozás feladata, hogy az adó ne adjon több adatot, mint amit a vevő fogadni tud. A multiplexelés lehetővé teszi, hogy több alkalmazás használja ugyanazt a fizikai összeköttetést. Az összeköttetések nyalábolása pedig lehetővé teszi, hogy egy alkalmazás használjon több összeköttetést egy időben.

3.5. Az együttműködési réteg

A réteg lehetővé teszi, hogy két számítógép felhasználói kapcsolatot létesítsenek egymással. A viszonyréteg segítségével egy felhasználó állományokat mozgathat számítógépek között. Jellegzetes feladata a logikai kapcsolat felépítése és bontása, párbeszéd szervezése. Szinkronizációs feladatokat is ellát, ellenőrzési pontok beépítésével.

3.6. A megjelenítési réteg

A réteg a viszonyrétegen felül helyezkedik el, és olyan szolgáltatásokat ad, amelyekre a legtöbb alkalmazói programnak szüksége van, amikor a hálózatot használja. Ez a réteg foglalkozik a hálózaton továbbítandó adatok ábrázolásával: el kell döntenie, hogy milyen egységes struktúrába szervezze az adatokat, amelyeket a felette elhelyezkedõ alkalmazói rétegtõl kap. A legtöbb program például neveket, számokat, stb. küld egymásnak, amelyeket esetenként bonyolult adatszerkezetekként ábrázolnak. Ehhez jön még az a tény, hogy a különböző számítógépek különböző kódolásokat alkalmaznak (ASCII,...). Annak érdekében, hogy a számítógépek egymással kommunikálni tudjanak, az adatokat a hálózaton egységes szabvány szerint kell bitek egymásutánjára kódolni. Ezt végzi el a megjelenítési réteg. Egyéb feladatai közé tartozhat még az adattömörítés, illetve a titkosítás is.

3.7. Az alkalmazási réteg

Széles körben igényelt protokollokat tartalmaz. Az állománytovábbításokon kívül ehhez a réteghez tartozik még az elektronikus levelezés, a távoli munkabevitel, a katalóguskikeresés, és még egy sor egyéb, általános-, ill. speciális célú alkalmazási feladat is. Ez a réteg kapcsolódik szorosan a felhasználóhoz, itt kell a hálózati felhasználói kapcsolatok megoldásait megvalósítani. Ezeken kívül definiál egy hálózati virtuális terminált amely segít az eltérő hálózati terminálok kezelésében.(Azok a szoftverek, melyeket már ténylegesen a felhasználó használ)

4. A TCP/IP

4.1. A protokoll

Az informatikában a protokoll egy egyezmény, vagy szabvány, amely leírja, hogy a hálózat résztvevői miképp tudnak egymással kommunikálni. Ez többnyire a kapcsolat felvételét, kommunikációt, adat továbbítást jelent. Gyakorlati szempontból a protokoll azt mondja meg, hogy milyen sorrendben milyen protokoll-üzeneteket küldhetnek egymásnak a csomópontok, illetve az üzenetek pontos felépítését, az abban szereplő adatok jelentését is megadja.

4.2. A port szám

Egy adott gépen több olyan program is futhat, amely hálózati kommunikációra képes. Ha egy gép „megszólít” egy másikat, a címzett honnét fogja

tudni, hogy mely programjával akarunk kommunikálni. A portszám valójában ezeket azonosítja. Képzeljünk el egy bankot, ahová egy ajtón lehet bejutni (Hálókártya), de benn több ablak van, számokkal megkülönböztetve, mindegyik mögött (már amelyik nyitva van) mással foglakozó munkatárs ül: pénztár, hitel, stb. Így van ez az informatikában is. Adjuk ki a következő utasítást! `cat /etc/services`
A kapott lista azokat a program - portszám összerendeléseket tatalmazza, amelyek a leggyakoribbak.

4.3. A TCP

Az üzenetek széttördelését, összeállítását, az elveszett részek újraadását, a datagrammok helyes sorrendjének visszaállítását mind a TCP (transmission control protocol – átvitelvezérlési protokoll) végzi. Az egyes datagrammok útvonalának a meghatározását (routing) az IP (internet protocol) hajtja végre. Mindez azt a látszatot kelti, hogy a munka tetemes része a TCP-re hárul. Kis kiterjedésû hálózatokban ez így is van, azonban az Interneten egy datagrammnak a rendeltetési helyre való juttatása igen összetett feladatot jelenthet. Egy datagramm több hálózaton mehet keresztül míg végül eljut a célállomásra. Például a Rutgers Egyetemrõl kiindulva a John von Neumann Supercomputing Center-ig soros vonalon keresztül, majd onnan (egy pár Ethernet hálózaton átjutva) 56Kbaud telefonvonalakon keresztül jut el egy másik NSFnet hálózatra stb... A különbözõ átviteli közegekbõl adódó inkompatibilitások kezelése és a célállomásokhoz vezetõ útvonalak végigkövetése komplex feladat. Meg kell jegyezni azonban, hogy a TCP és az IP közti interfész rendkívül egyszerû: a TCP egy datagrammot ad át az IP-nek egy rendeltetési címmel együtt. Az IP semmit sem tud arról, hogy ez az információ hogyan viszonyul más datagrammokhoz.
Az alábbiakban a tipikus TCP/IP hálózaton keresztül haladó üzenetre rárakódó fejléceket tekintjük át:
Kezdetnek vegyünk egy egyszerû adatfolyamot (pl. egy állomány tartalma), amelyet egy másik számítógépnek szeretnénk elküldeni:

.....

Ezt a TCP megcsonkítja. (Ennek érdekében tudatni kell a protokollal, hogy mekkora az a maximális adatméret, amelyet az adott hálózat még kezelni tud. Valójában az összeköttetés két végén a TCP-k közlik egymással az általuk kezelhetõ maximális méretet, majd veszik a kisebbiket.)

....

Minden datagramm elé egy TCP fejléc kerül, amely legalább 20 oktetből áll. Ezek közül a legfontosabbak: egy forrás- és egy célport, valamint egy sorszám. A portok az összeköttetések végpontjait azonosítják. Tegyük fel például, hogy egyszerre 3 felhasználó továbbít állományokat. A TCP ezekhez az átvitelekhez az 1000, 1001 és 1002 portokat rendelheti. Datagramm küldésekor az allokált port válik a forrásporttá, mivel innen indul ki a datagramm. A kapcsolat másik végénél lévõ TCP szintén hozzárendeli a saját portját az átvitelhez. A küldõ oldali TCP-nek a célport számát is tudnia kell (ezt az információt a kapcsolat felépülésekor szerzi meg), amelyet az a fejléc célport mezőjébe helyez. Ha a másik oldalról érkezik egy datagramm, akkor annak TCP fejlécében a forrás- és a célportok tartalma ellentétes, hiszen ekkor az a forrás, ez pedig a rendeltetési hely. Minden datagrammnak van egy sorszáma, amely a vevő oldalt arról biztosítja, hogy minden adatot helyes sorrendben kapjon meg, és ne veszítsen el egyet se a datagrammok közül.
Ha a TCP fejlécet T-vel jelöljük, akkor az eredeti állományunk így néz ki:
T.... T.... T.... T.... T.... T.... T.... T.... T....

4.4. Az IP

A TCP az általa feldolgozott datagrammokat átadja az IP-nek. Persze ezzel együtt közölnie kell a rendeltetési hely Internet címét is. Az IP-t ezeken kívül nem érdekli más: nem számít, hogy mi található a datagrammban vagy, hogy hogyan néz ki a TCP fejléc. Az IP feladata abban áll, hogy a datagramm számára megkeresse a megfelelõ útvonalat és azt a másik oldalhoz eljuttassa. Az útközben fellelhetõ átjárók és egyéb közbülsõ rendszereken való átjutás megkönnyítésére az IP a datagrammhoz hozzáteszi a saját fejlécét. A fejléc fõ részei a forrás, és a rendeltetési hely IP címe, a protokollszám és egy ellenőrző összeg. A forrás címe a küldő gép címét tartalmazza. (Ez azért szükséges, hogy a vevő oldal tudja honnan érkezett az adat.) A rendeltetési hely címe a vevő oldali gép címét jelenti. (Ez pedig azért szükséges, hogy a közbenső átjárók továbbítani tudják az adatot.) A protokollszám kijelöli, hogy a datagramm a különböző szállítási folyamatok közül melyikhez tartozik.
Ha az IP fejlécet I-vel jelöljük, akkor az eredeti állományunk így néz ki:
IT.... IT.... IT.... IT.... IT.... IT.... IT.... IT.... IT....
Ha ethernet hálózatunk van, akkor erre még az ethernet is ráülteti a maga fejlécét, a végére illeszt egy ellenőrző összeget (a hibamentes átvitelt lehet ezzel ellenőrizni). Ha az Ethernet fejlécet E-vel, az ellenõrzõösszeget pedig C-vel jelöljük, akkor az eredeti állományunk így néz ki:
EIT....C EIT....C EIT....C EIT....C EIT....C EIT....C EIT....C

A csomagok megérkezésekor persze a fenti fejlécek mindegyikét leszedi a megfelelõ protokoll. Az ethernet interfész az ethernet fejlécet és az ethernet ellenőrző összeget szedi le. Ezekután ellenőrzi a típuskódot. Mivel az az IP-re mutat, ezért a datagrammot átadja az IP-nek, amely a Protokoll mező tartalmát ellenőrzi. Itt azt találja, hogy TCP, ezért a datagrammot a TCP-nek adja át. A TCP a Sorszám mező tartalma és egyéb információk alapján állítja össze az eredeti állományt.

5. Egyéb fontos protokollok

–UDP: A User Datagram Protocol (UDP) az internet egyik alapprotokollja. Feladata datagram alapú szolgáltatás biztosítása, azaz rövid, gyors üzenetek küldése. Jellemzően akkor használják, amikor a gyorsaság fontosabb a megbízhatóságnál, mert az UDP nem garantálja a csomag megérkezését. Ilyen szolgáltatások például a DNS, a valós idejű multimédia átvitelek, vagy a hálózati játékok.
–NTP: Network Time Protokol. Az idő hálózaton keresztüli szinkronizálását végző protokol.
–FTP: File Transfer Protocol. Fájlok le illetve feltöltésére használt protokol.
–POP3: Post Office Protokol. Levelezésnél levelek olvasásához használható protokol, a levelet letölti a levelező kliens a szerverről a helyi meghajtóra.
–IMAP: Internet Message Access Protocol. A POP3 rokona, a leveleink a levelező szerveren maradnak. Sok levelezéshez kapcsolódó szolgáltatást nyújt.
–HTTP: Hyper Text Transfer Protokol. A HTTP egy kérés-válasz alapú protokoll kliens és szerver között.

Tartalomjegyzék