BEVEZETÉS AZ INTERNET PROTOKOLLOKBA
Computer Science Facilities Group
RUTGERS
New Jersey Állami Egyetem
1996. Június

Ez a dokumentum egyfajta bevezetésként szolgál az Internet hálózati protokollokba (TCP/IP). Összegzi az elérhető szolgáltatásokat, illetve a főbb protokollok rövid leírását tartalmazza.

Tartalomjegyzék

MEGJEGYZÉS

Próbáltam a magyar nyelvű szakirodalomban többé-kevésbé elterjedt kifejezéseket használni. Ha egy kifejezés kapcsán úgy éreztem, hogy arra nincs igazán megfelelő, széles körben elterjedt változat, akkor megtartottam az angol nyelvűt.

Ez a magyar nyelvű változat egyben aktualizált, frissített változata az eredeti angol nyelvű szövegnek. Az írása óta eltelt idő alatt elég jelentős változások mentek végbe a hálózatok fejlődése terén. Ugyanakkor a dokumentum általános voltának köszönhetően a lényegi részek nem, vagy csak alig változtak.

Budapest, 1996. június 10.

Vincze Tamás
tvs@maxi.inf.elte.hu

A jelen dokumentum a TCP/IP protokollokba vezet be röviden, majd ötleteket ad a további információk gyűjtéséhez. Semmi esetre sem kíván teljes értékű leírás lenni, csupán csak a protokollok ötletvilágát tárja elénk.

Amennyiben további műszaki kérdések érdeklik, olvassa el magukat a szabványokat. A szöveg tele van ezekre vonatkozó hivatkozásokkal, úgynevezett RFC és IEN számok formájában, amelyek dokumentumokat jelölnek.

Az utolsó fejezet foglalkozik azzal, hogy ezek a számok mit jelentenek és hogyan lehet előcsalogatni belőlük a hozzájuk tartozó szabványokat, hogyan lehet másolatot szerezni.

1. Mi is az a TCP/IP?

A TCP/IP nem más, mint egy protokollkészlet, amelyet arra dolgoztak ki, hogy hálózatba kapcsolt számítógépek megoszthassák egymás között az erőforrásaikat. A fejlesztés az ARPAnet köré csoportosult kutatók munkája.

Valószínűleg az ARPAnet a legismertebb TCP/IP alapú hálózat.

Hedrick azt írja, hogy „1987 júniusáig legalább 130 különböző cég adott ki olyan terméket, amely a TCP/IP-t támogatta, és többezer hálózat alkalmazza is a protokollokat”. Én nem jártam utána, hogy ma mekkora lehet ez a szám, de nyilvánvalóan ennél sokkal több. 1987 óta az Internet jelentősen meghízott.

Először tekintsük át az alapvető fogalmakat. Az itt leírt protokollkészlet legjobb elnevezése „Internet protokollverem” (vagy Internet protokollkészlet). A TCP és az IP ezen protokollok közül kettő. (Leírásukat lásd lejjebb.) Mivel a protokollok közül a TCP és az IP a legismertebb, ezért az egész családra a TCP/IP vagy az IP/TCP kifejezést használják. Valószínűleg nincs is értelme ellenkezni. Ennek következtében jónéhány furcsasággal találhatjuk szembe magunkat. Egyszer például azon kaptam magamat, hogy az NFS-ről úgy beszéltem, mintha az TCP/IP alapú lenne, miközben az egyáltalán nem is használ TCP-t. (IP-t viszont igen. A TCP helyett egy másik protokollt, UDP-t használ. A sok rövidítésről a következő oldalakon lelebbentjük a fátylat.)

Az Internet: hálózatok együttese. Hozzátartozik az Arpanet, az NSFnet, regionális hálózatok (mint a NYsernet), számos egyetem és kutatóintézet helyi hálózata, és egy sor katonai hálózat is. Az „Internet” kifejezés ezen hálózatoknak az összességét jelenti. Ennek egy része a DDN (Defense Data Network), amely az USA Védelmi Minisztériumának az irányítása alatt áll.

Ide tartozik néhány kutatói hálózat (pl. Arpanet), illetve sokkal titkosabb katonai hálózatok is. (Mivel az Internet protokollok fejlesztéséhez való anyagi hozzájárulások nagy része DDN szervezetektől származik, ezért az Internet és a DDN kifejezések néha egybemosódni látszanak.) A fenti hálózatok mindegyike összeköttetésben áll egymással. A felhasználók bármelyikről bármelyikre küldhetnek üzenetet, kivéve azokat, ahol biztonsági vagy egyéb okokból megszorították a hozzáférést. Az Internet protokollokat leíró dokumentumok olyan hivatalos szabványok, amelyeket az Internetet használók közössége elfogadott és alkalmaz. Az USA Védelmi Minisztériuma 1987 tájékán kiadta a TCP/IP MILSPEC-féle definícióját. A helyzet az, hogy a TCP/IP hívők továbbra is az Internet szabványokat használják. A MILSPEC változat azokkal konzisztens.

Mindegy, hogy minek nevezzük, a TCP/IP egy protokollcsalád. Jónéhány tagja biztosít sok alkalmazás számára szükséges alacsony szintű szolgálatokat. Ilyen például az IP, a TCP és az UDP. (Ezeket egy kicsit később részletesebben is megnézzük.) Mások olyan meghatározott feladatokat látnak el, mint például a számítógépek közötti állománytovábbítás, az üzenetküldés, vagy éppen egy adott gépre bejelentkezett felhasználók lekérdezése. A TCP/IP-t kezdetben főleg kis- és nagyszámítógépek (mainframe-ek) körében alkalmazták. Ezek a gépek saját merevlemezzel rendelkeztek, és általában teljesen önállóak voltak. Innen származtathatók a TCP/IP legfontosabb „hagyományos” szolgáltatások:

- **állománytovábbítás.** Az állománytovábbítási protokoll (File Transfer Protocol, azaz FTP) segítségével bármely számítógépen lévő felhasználó bármelyik másik gépre küldhet és onnan beszerezhet állományokat. A biztonságot a felhasználónak a másik gépen érvényes azonosítója és a hozzátartozó jelszava jelenti. Gondoskodtak arról is, hogy a különböző karakterkészlettel, sorvégjellel stb... rendelkező számítógépek közötti állománytovábbítás is zavartalan legyen. Ez nem teljesen ugyanaz a dolog mint a hálózati állományrendszer (network file system) vagy a netbios protokoll, amelyekről később lesz szó. Az FTP egy olyan segédprogram, amelyet bármely időpontban futtatva, a hálózatba kapcsolt más számítógépeken lévő állományok elérhetővé válnak. Arra használják, hogy az adatállományt a saját rendszerre átmásolják. (Az FTP leírását lásd az RFC 959-ben).
- **távoli bejelentkezés.** A hálózati terminál protokoll (TELNET) a felhasználók távoli gépekre való bejelentkezését kezeli. A távoli viszonyt (session) annak a gépnek a megadásával kell kezdeni, amelyhez csatlakozni szeretnénk. Attól kezdve bármit is gépelünk be, minden adat a megadott géphez kerül a viszony befejeztéig. Vegyük észre, hogy a felhasználó valójában még mindig a saját számítógépével kommunikál. A telnet program az, amelyik a futása alatt ezt láthatatlanná teszi a felhasználó előtt. Minden begépelt karakter közvetlenül a másik rendszerhez kerül. A távoli géppel meglévő kapcsolat nagyjából hasonlít egy modemes

vonalhoz (dial-up connection). Ez azt jelenti, hogy a távoli rendszer először a bejelentkezést kéri, majd egy jelszót, ugyanúgy, ahogy ez egy modemes kapcsolat esetén történne. A kijelentkezéskor a telnet program kilép a vonalból, és ismét a saját gépünk kommunikál velünk. A telnet program kisszámítógépes megvalósításai általában egy elterjedt terminháltípus emulációját is tartalmazzák. (A specifikációt lásd az RFC 854 és az RFC 855 dokumentumokban. Az RFC 854 -- 860 a TELNET opcióit írja le. Hamár itt tartunk, ne tévesszük össze a telnet protokollt a Telenet-tel, amely egy hálózati szolgáltatásokat kínáló kereskedelmi cég neve.)

- **számítógépes levelezés (mail).** Ez a szolgáltatás arra való, hogy a felhasználók üzeneteket küldjenek egymásnak. Az emberek kezdetben csak egy-két számítógépet használtak. Ezeken a gépeken aztán mindenki a saját levelezési állományát tartotta fenn. Levél, illetve üzenet elküldésekor annyi történik, hogy az egyszerűen a címzett megfelelő állományához fűződik. Az olyan környezetben azonban, ahol mikroszámítógépeket használnak, ezzel gond van. A legalapvetőbb probléma abból fakad, hogy a mikroszámítógépek nem a legmegfelelőbbek üzenetek fogadására. Levél küldésekor a levelezést végző program kommunikációs csatornát akar megnyitni a címzett géppel. Ha ez a gép történetesen egy mikroszámítógép, akkor lehetséges, hogy éppen ki van kapcsolva, vagy esetleg nem az üzeneteket kezelő alkalmazást futtatja. Ennek a problémának a megoldására az üzeneteket kezelését egy állandóan futó kiszolgáló (mail server) végzi el. A mikrogépeken futó levelező program pedig egy felhasználói interfészt alkot a kiszolgáló felé. (Lásd az RFC 821 és 822 dokumentumokat a számítógépes levelezésre vonatkozólag. Az RFC 937 pedig egy olyan protokollt ír le, amely a mikroszámítógépeknek a levelezést kiszolgáló számítógéptől való üzenetfogadását specifikálja.)

A TCP/IP protokollok bármely megvalósításának tartalmaznia kell a fenti szolgáltatások mindegyikét. A mikroszámítógépes implementációkban a levelező rendszer nem mindig szerepel. Ezek a megszokott, hagyományos alkalmazások fontos szerepet játszanak a TCP/IP alapú hálózatokban. A hálózatokról alkotott elképzelés azonban folyamatosan változik. Manapság már sok helyen többfajta számítógép is működik egyszerre: mikroszámítógépek, munkaállomások, kisszámítógépek, illetve nagyteljesítményű számítógépek. Ezen gépek mindegyikét speciális feladatokra állították fel. Habár az emberek többsége még mindig csak egy meghatározott számítógépet használ a munkája során, ez a gép a kívánt szolgáltatások eléréséhez egyéb hálózati erőforrásokat vesz igénybe. Ez a modell hozta létre a „server/client” (kiszolgáló/kliens) alapú hálózati szolgáltatásokat. A kiszolgáló nem más mint egy meghatározott hálózati rendszer, amely a hálózat többi tagja részére biztosít bizonyos szolgáltatásokat. A kliens pedig az a rendszer, amely a szolgáltatást igénybe veszi. (Nem szükséges, hogy a kiszolgáló és a kliens különböző számítógépen legyen. Lehetnek például egyazon a számítógépen futó különböző programok is.) Az alábbiakban felsoroljuk a mai hálózati felépítésben jelenlévő tipikus kiszolgálókat. Ezek a szolgáltatások a TCP/IP keretén belül is megtalálhatók.

- **hálózati állományrendszer (network file system).** Ennek a szolgáltatásnak a segítségével a hálózaton lévő állományokat az FTP módszerénél valamivel természetesebben lehet elérni. A hálózati állományrendszer azt az illúziót kelti, hogy az egyik rendszer lemezei vagy más egységei közvetlenül más rendszerekhez tartoznak. Nincs szükség külön hálózati alkalmazásra ahhoz, hogy az állományokhoz hozzá lehessen férni. Az adott számítógép egyszerűen úgy viselkedik, mintha plusz egységeket kapott volna. Ezek a „virtuális” meghajtók a másik rendszer lemezeit fogják jelenteni. Több hasznos oldala is van ennek a megközelítésnek. Egyfelől nagykapacitású meghajtókat lehet megosztani több számítógép között. Ennek nyilvánvaló takarékossági előnyei vannak. Másfelől egy csapásra megvalósul a közös állomány-hozzáférés. Könnyebbé válik a rendszer karbantartása, archiválása, mivel nem kell a különböző gépeken lévő másolatok időszerűsítésével és tartalékolásával foglalkozni. Sok cég kínál nagyteljesítményű, meghajtó nélküli számítógépeket. Ezeknek a gépeknek a működése nagy mértékben a különböző állománykiszolgálókhoz kapcsolt meghajtóktól függ. (A TCP alapú, PC-re készült NetBIOS leírását az RFC 1001 és 1002 adja. A munkaállomások és a kisszámítógépek körében a Sun Network File System az irányadó. Ennek a protokoll specifikációit a Sun Microsystems cég szolgáltatja.)

- **távoli nyomtatás.** Itt arról van szó, hogy a más számítógépekhez csatlakoztatott nyomtatókat sajátként tudjuk elérni. (A legszélesebb körben használt protokoll a Berkeley Unix távoli sornyomtatás protokollja. Sajnos ennek dokumentált verziója nem létezik. A C nyelvű forráskódot a Berkeley egyetemtől könnyen be lehet szerezni, ami a megvalósítást könnyíti.)

- **távoli futtatás.** A szolgáltatás megengedi programok másik gépen való végvégrehajtását. Ez akkor hasznos, ha a munka nagy részét kisebb teljesítményű gépen el lehet végezni, de néhány feladat nagyobb rendszer erőforrásait igényli. A távoli futtatásnak jónéhány fajtája létezik. Vannak olyanok, amelyek a parancsokat parancs szinten hajtják végre. (Az intelligensebb változatok olyan rendszert keresnek, amely szabad erőforrással rendelkezik). Léteznek távoli eljáráshívó rendszerek is, amelyek megengedik, hogy a programok másik gépen futó szubrutinokat hívjanak meg. (Ennek a megvalósítására több protokoll is létezik. A Berkeley Unix-ban két kiszolgáló is található a távoli futtatásra: az rsh és az rexec. Ezek man oldalai írják le az általuk használt protokollokat. A Berkeley 4.3 verzióval terjesztett programcsomag tartalmaz egy olyan osztott parancsértelmezőt, amely a rendszer terhelésétől függő mértékben osztja fel a feladatokat különböző rendszerek között. A távoli eljáráshívások módszere az 1980-as évek vége felé a

kutatások középpontjában állt, aminek eredményeképpen sok szervezet rendelkezik a szolgáltatás implementációjával. A leginkább elterjedt és kereskedelmileg is támogatott távoli eljáráshívó protokoll a Xerox cég Courier, illetve a Sun cég RPC protokollja. A dokumentációk beszerezhetők maguktól a cégektől. A Courier-nek létezik TCP alapú, publikus megvalósítása is, amelyet a Berkeley 4.3 programcsomag részeként terjesztenek. Az RPC egy Sun megvalósítása a Usenet hálózaton volt megtalálható, illetve a Berkeley 4.3 részeként is megjelent.)

- **névkiszolgálók (name servers).** Nagy kiterjedésű rendszerek működése során rengeteg név keletkezik, amit valahogy adminisztrálni kell. Ide tartoznak a felhasználók és a jelszavaik, az azonosítók és a számítógépek nevei és hálózati címei. Ha mindezeket minden számítógépen naprakészen akarnánk tartani, akkor elvesznénk az információ dzsungelében. Ennek elkerülése végett az adatbázisokat nem mindegyik, hanem csak egy pár rendszeren tartják fenn. A többi rendszer az adatokhoz a hálózaton keresztül fér hozzá. (Az Interneten lévő gépek neveit és Internet címeit nyomon követő névkiszolgáló protokollokat az RFC 822 és 823 dokumentumok írják le. Ez ma már bármely TCP/IP megvalósításnak része kell, hogy legyen. Az IEN 116 egy olyan régebbi névszolgáltató protokollt ír le, amelyet még egy pár terminálkiszolgáló és egyéb termék használ a számítógépek keresésére. A Sun cég Yellow Pages rendszere a felhasználók neveinek, az állomány-megosztó csoportoknak, illetve a Unix rendszerek által használt más adatbázisoknak az általános kezelésére szolgál. A rendszer a kereskedelemben kapható. A protokoll leírása a Sun cégtől szerezhető be.)

- **terminálszerverek.** Rengeteg rendszerben előfordul, hogy a terminálokat nem csatlakoztatják közvetlenül a számítógépekhez. Ehelyett ezek úgynevezett terminálszerverekhez csatlakoznak. A terminálszerver nem más mint egy kisteljesítményű számítógép, amely csak a telnet (vagy más, bejelentkezést végrehajtó protokoll) futtatására hivatott. Amennyiben a használt terminál ilyen számítógéphez van kötve, akkor egyszerűen csak be kell gépelni egy számítógép nevét, és máris létrejön a kapcsolat. Általában lehetséges egyszerre több számítógép felé aktív kapcsolat fenntartása is. A terminálszerver végzi az élő kapcsolatok közötti váltogatást, és figyelmezteti a felhasználót, ha egy kapcsolat kimenetén megjelenik valami. (A terminálszerverek a már említett telnet protokollt használják. A valódi terminálszervereknek tudniuk kell a névszolgálatot, illetve egyéb protokollokat is.)

- **hálózat alapú ablakos rendszerek.** A nagy teljesítményű grafikai programok régebben olyan számítógépeket igényeltek, amelyekhez közvetlenül csatlakozott bittérképes grafikus képernyő. A hálózati ablakos rendszerek megengedik, hogy az ilyen programok más számítógéphez csatlakoztatott kijelzőt használjanak. Ezek a rendszerek biztosítják, hogy a különböző feladatokat a legmegfelelőbb rendszerek végezzék, miközben végig egyetlen grafikus felületet mutatnak a felhasználó felé. (A legelterjedtebb megvalósítás az X. A protokoll leírását a MIT Athena projektjétől lehet beszerezni. Sok cég támogatja a Sun NeWS nevű rendszerét is. Mindkét rendszer TCP/IP-re épül.)

A fenti protokollok közül jónéhány a Sun, a Berkeley, illetve más szervezetek munkájának eredménye. Ez azt jelenti, hogy ezek hivatalosan nem részei az Internet protokollkészletnek; persze a megvalósításukban TCP/IP-t használnak, mint bármely más TCP/IP alkalmazás. Mivel a protokoll definíciók nem képeznek tulajdonjogot, valamint mivel kereskedelmileg támogatott megvalósítások is hozzáférhetőek, ezért célszerű ezeket a protokollokat az Internet protokollkészlet részeként tekinteni. A fenti lista a TCP/IP-n keresztül elérhető szolgáltatásokból csak egy mintafelsorolás, a főbb alkalmazások többségét azonban tartalmazza. A többi széles körben használatos protokoll olyan speciális információkat biztosít, mint például a bejelentekezett felhasználók, az aktuális idő stb... Amennyiben olyan szolgáltatásra van szükség, amely nem szerepel a fentiek között, akkor ajánlott az Internet Protokollok aktuális listájának (RFC 1011) a megtekintése. Ebben megtalálható minden protokoll. Érdemes még a főbb TCP/IP megvalósításokat is végigböngészni, hogy a különböző cégek milyen újabb szolgáltatásokat adtak hozzá a protokollokhoz.

2. A TCP/IP protokollok általános jellemzői

A TCP/IP protokollkészlet egymásra épülő rétegekből áll. Ennek megértéséhez tekintsünk egy példát. Tipikus hálózati feladat a levelezés megvalósítása, amit protokoll szabályoz. A protokoll az egyik gép által a másiknak küldendő parancsokat definiálja, például annak meghatározására, hogy ki a levél küldője, ki a címzett, majd ezután következik a levél szövege. A protokoll feltételezi továbbá, hogy a kérdéses két számítógép között megbízható kommunikációs csatorna létezik. A levelezés, mint bármely más alkalmazási rétegbeli protokoll, a küldendő parancsokat és üzeneteket definiálja. A tervezésekor a TCP/IP-t vették alapul, azaz azzal együtt használható. A TCP a felelős azért, hogy a parancsok biztosan elkerüljenek a címzetthez. Figyel arra, hogy mi került át, és ami nem jutott el a címzetthez, azt újraadja. Amennyiben egy falat, pl. az üzenet szövege, túl nagy lenne (meghaladja egy datagramm méretét), akkor azt a TCP széttördeli több datagrammra, és biztosítja, hogy azok helyesen érkezzen célba. Mivel a fenti szolgáltatásokat jónéhány alkalmazás igényli, ezért ezeket nem a levelezés, hanem egy külön protokoll tartalmazza. Az egész TCP tulajdonképpen nem más, mint rutinok olyan gyűjteménye, amelyet a különböző alkalmazások vesznek igénybe, hogy megbízható hálózati kapcsolatot építsenek ki más számítógépekkel. A TCP hasonlóképpen alapul az IP szolgáltatásokon. Habár a TCP

szolgáltatásait sok alkalmazás igényli, vannak olyanok, amelyeknek nincs rájuk szükségük. Persze léteznek olyan szolgáltatások, amelyeket minden alkalmazás megkíván. Ezeket szedték egybe az IP-be. Ugyanúgy, ahogy a TCP, az IP is egy rutingyűjtemény, de ezt a TCP-t nem használó alkalmazások is elérhetik. A különböző protokolloknak ezt a szintekbe rendezését rétegezésnek nevezik. Ennek megfelelően az alkalmazási programok (mint például a levelezés), a TCP, illetve az IP külön réteget alkotnak, amelyek mindegyike az alatta lévő réteg szolgáltatásait használja.

A TCP/IP alkalmazások általában a következő négy réteget veszik igénybe:

- alkalmazási protokollok (pl. levelezés);
- a TCP-hez hasonló protokollok, amelyek rengeteg alkalmazás számára biztosítanak szolgáltatásokat;
- IP, amely a datagrammok célba juttatását biztosítja;
- a felhasznált fizikai eszközök kezeléséhez szükséges protokollok (pl. Ethernet)

A TCP/IP alapjául az ún. „catenet" modell szolgált (részletesebben lásd: IEN 48). Az alapfeltevés az, hogy nagyszámú különböző hálózat áll egymással összeköttetésben átjárók segítségével. Ezeken a hálózatokon lévő bármely számítógépet vagy erőforrást a felhasználónak el kell tudnia érni. Az adatcsomagok esetleg több tucat hálózaton is keresztülmehetnek mielőtt a célállomásra érkeznének. Az ezt megvalósító útvonal-választásnak természetesen láthatatlannak kell maradnia a felhasználó számára, abból ő mindössze egy Internet címet kell, hogy ismerjen. Ez egy olyan adat, mint például a 128.6.4.194, ami tulajdonképpen egy 32 bites számot reprezentál. A felírás 4 darab 8 bites decimális szám formájában történik. (Az Internet dokumentációkban a byte helyett az oktet kifejezést használják a 8 bites számokra. Ez azért van így, mert a TCP/IP-t olyan számítógépek is használják, amelyek architektúrájában a byte nem 8 bites számot jelöl.) A cím alapján kideríthető, hogy hogyan lehet a rendszerhez eljutni (általában ez is a cím szerepe :-)). A fenti példában a 128.6 egy olyan hálózati szám, amelyet egy központi hatóság adott ki a Rutgers Egyetem számára. Az egyetem a következő oktetet a tanszékek azonosítására használja. A 128.6.4 az egyetem számítógéptudományi tanszékét jelöli. A negyedik, egyben az utolsó oktet maximum 254 rendszert azonosíthat minden esetben (azért 254, mert a 0 és a 255 nem megengedett értékek; erről később lesz szó). A fentiek szerint a 128.6.4.194. és a 128.6.5.194 nem ugyanaz a rendszer. Az Internet cím szerkezetéről bővebben lásd később.

A különböző rendszerekre általában a nevükkel hivatkozunk, és nem az Internet címükkel. Egy ilyen név megadásakor a hálózati szoftver egy adatbázisból kikeresi a hozzátartozó címet. Ez azért fontos, mert a legtöbb hálózati szoftver címekkel operál. (A keresés-hozzárendelés leírását lásd az RFC 882-ben.)

A TCP/IP összeköttetés-mentes hálózati protokollokat tartalmaz, ami azt jelenti, hogy az információ a datagrammok sorozataként terjed tovább. A datagramm adatok együttese, amely egy egyszerű üzenetként kerül továbbításra. A datagrammok egymástól függetlenül, egyesével indulnak útjukra. (Az adott adatkapcsolat időtartamára vonatkozóan persze vannak előrejelzések.) A küldendő információt egy meghatározott szinten a protokollok a fenti adatokra tördelik, amelyeket aztán a hálózat egymástól teljesen különállóként kezel. Tegyük fel például, hogy egy 15000 oktet méretű állomány továbbításáról van szó. Mivel a legtöbb hálózat nem tud ekkora datagrammal mit kezdeni, ezért azt a protokollok mondjuk 30 darab 500 oktetes darabra szedik szét, amelyek mindegyikét elküldik a célállomásra. Ott aztán belőlük összerakják az eredeti 15000 oktetes állományt. A datagrammok adása közben a hálózaton semmi nem utal arra, hogy közöttük bármiféle kapcsolat is létezne; előfordulhat, hogy egy a sorrendben eredetileg hátrább álló megelőz egy előtte állót. Az is lehetséges, hogy a hálózaton valahol hiba keletkezik és néhányuk nem érkezik meg a rendeltetési helyére. Ilyenkor újra kell adni a hiányzó datagrammot.

A datagramm és a csomag kifejezés gyakran egymással felcserélhetőnek tűnik, azonban ez nem minden esetben van így. A TCP/IP leírásakor a datagramm a helyes kifejezés: azt az adategységet jelöli, amellyel a protokollok operálnak, míg a csomag egy fizikailag létező dolog, amely a kábeleken jelenik meg. A legtöbb esetben egy csomag egyetlen datagrammot tartalmaz. Ilyenkor szinte elenyésző a különbség. Vannak azonban kivételek: X.25 kábelezésre épülő TCP/IP esetén a két réteg közötti X.25 interfész a datagrammokat 128 byte-os csomagokra tördeli. Mindezt a TCP/IP természetesen nem veszi észre, hiszen a célállomáson a csomagokat ismét egyetlen datagrammá rakja össze az interfész. Itt tehát egyetlen datagrammot több különböző csomag szállít. A legtöbb közegben ezek a különbségek egyre inkább eltűnni látszanak.

2.1 A TCP szint

A TCP/IP datagrammok kezelésében két különböző protokoll játszik szerepet. 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 kiterjedésű 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 Egyetemről 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 különböző átviteli közegekből adódó inkompatibilitások kezelése és a célállomásokhoz vezető útvonalak végigkövetése komplex feladat. Meg kell jegyezni azonban, hogy a TCP és az IP közti interfész rendkívül egyszerű: 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 olvasásban idáig eljutva felmerülhet a gyanú, hogy az eddig elmondottak nem alkotnak egészen teljes keretet. Szó volt ugyan az Internet címekről, de arról nem: vajon hogyan lehet egy adott rendszer esetén az ahhoz befutó különböző kapcsolatokat nyomon követni? Nyilván nem elegendő csak a datagrammnak a helyes címre való továbbítása. A TCP-nek még azt is tudnia kell, hogy az adott datagramm melyik kapcsolathoz tartozik. A probléma megoldását a demultiplexálás v. nyalábbontás néven ismert eljárás adja, amely a TCP/IP-ben valójában több különböző szinten folyik. A demultiplexáláshoz szükséges információt az ún. fejlécek hordozzák. A fejléc azokat az extra okteteket jelenti, amelyeket a különböző protokollok ragasztanak a datagrammok elejére, hogy azokat nyomon tudják követni. A dolog hasonlít ahhoz, amikor a levelet a borítékba tesszük, majd azt megcímezzük. A különbség annyi, hogy a modern hálózatokban ez jóval többször történik: olyan mintha a levelet egy kis borítékba tennénk, majd azt a titkárnőnk egy nagyobb borítékba helyezné, amit a központ egy még nagyobb borítékban továbbítana stb... Az alábbiakban a tipikus TCP/IP hálózaton keresztül haladó üzenetre rárakódó fejléceket tekintjük át:

Kezdetnek vegyünk egy egyszerű 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 kezelhető 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 lévő TCP szintén hozzárendeli a saját portját az átvitelhez. A küldő oldali TCP-nek a célport számát is tudnia kell (ezt az információt a kapcsolat felépülésekor szerzi meg; lásd lejjebb), 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. (További részleteket illetően lásd a TCP specifikációkat.) A TCP valójában nem a datagrammokat, hanem az okteteket sorszámozza. Ha például minden datagramm 500 oktet adatot tartalmaz, akkor az első datagramm sorszáma 0, a másodiké 500, a következőé 1000, az az utánié 1500 stb... lesz. Végül essék szó az ellenőrzőösszegről: ez egy olyan szám, amelyet a datagrammban lévő oktetek összeadásával kapunk (többé-kevésbé; lásd a TCP specifikációt. Az OSI szállítási rétege ezt úgy képzi, hogy az adatokat 16 bites számokként összeadja, majd veszi ennek egyes komplemensét—a fordító.) Az eredmény aztán bekerül a TCP fejlécbe. A vevő oldali TCP is kiszámítja a fenti algoritmus szerinti ellenőrzőösszeget. Ha a kettő nem egyezik, akkor a datagrammal az átvitel közben valahol valami baj történt és azt a protokoll eldobja. A datagramm mostanra tehát így néz ki:

```
<<-------------------------  32 bit  -------------------------->>
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Forrásport           |             Célport           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Sorszám                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Ráültetett nyugta                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TCP   |  Fenn-    |U|A|P|R|S|F|                                 |
|fejrész|  tartott  |R|C|S|S|Y|I|            Ablak               |
|hossza |           |G|K|H|T|N|N|                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Ellenőrzőösszeg        |       Sürgősségi mutató       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     tényleges adatok ... következő 500 oktet
|     .....
```

Ha a TCP fejlécet T-vel jelöljük, akkor az eredeti állományunk így néz ki:

```
T....  T....  T....  T....  T....
```

A fejlécben vannak olyan mezők, amelyekről még nem esett szó. A legtöbbjük az összeköttetés menedzselésével kapcsolatos információkat hordozza. A datagrammnak a rendeltetési helyre való megérkezését a vevő egy nyugtával hozza a küldő oldal tudomására. Ez a szám a datagramm TCP fejlécében a Ráültetett nyugta mezőben jelenik meg. Például egy olyan csomag elküldése, amelynek nyugtamezőjében 1500 szerepel, azt jelenti, hogy az 1500-as oktetig bezárólag minden datagramm eljutott a rendeltetési helyre. Amennyiben a küldő oldal egy adott időn belül nem kap nyugtát, akkor újból elküldi az adatot. Az Ablak mezőben lévő érték az összeköttetés alatt forgalomban lévő adatok mennyiségét határozza meg. Nem lenne szerencsés, ha minden egyes datagramm elküldése előtt meg kellene várni az előző nyugtáját, mert így a forgalom rendkívüli mértékben lelassulna. Másrészt viszont nem lehet folytonosan küldeni az adatokat, hiszen például egy gyorsabb számítógép adatárama elárasztaná a lassabb gépeket. Ennek megoldására mindkét oldal az Ablak mezőben elhelyezett oktetek számával közli, hogy éppen mekkora adatmennyiséget képes még befogadni. Az adatok vételével ez a szám, azaz az ablak mérete, folyamatosan csökken. Amikor eléri a nullát a küldőnek szüneteltetnie kell az adatok továbbítását. A vevő ablakmérete az adatok feldolgozása során nő, ami jelzi, hogy kész további adatok fogadására. Gyakran ugyanaz a datagramm használható az újabb adatok engedélyezésére és nyugtázásra is (aktualizált ablak segítségével). A Sürgősségi mutató mezőben lévő érték beállításával bármelyik oldal utasíthatja a másikat arra, hogy a feldolgozást egy adott oktettel folytassa. A gyakorlatban többek között ez az aszinkron eseményekkel kapcsolatban használatos, például amikor vezérlőkarakter vagy más, a kimenetet megszakító parancs kerül továbbításra. A többi mezőről ez a dokumentum nem hivatott szólni.

2.2 Az IP szint

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 megfelelő útvonalat és azt a másik oldalhoz eljuttassa. Az útközben fellelhető átjárók és egyéb közbülső rendszereken való átjutás megkönnyítésére az IP a datagrammhoz hozzáteszi a saját fejlécét. A fejléc fő részei a forrás, és a rendeltetési hely Internet címe (32 bites címek, pl. 128.6.4.94), 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. A TCP egy biztos választási lehetőség, de léteznek egyebek is (pl. UDP). Végül az ellenőrzőösszeg segítségével bizonyosodik meg a vevő oldali IP arról, hogy a fejléc az átvitel során nem sérült-e meg. A TCP és az IP különböző ellenőrzőösszegeket használ. Az IP-nek meg kell tudnia győződni a fejléc sértetlenségéről, különben rossz helyre küldhet el adatot. A TCP és az IP a biztonság és a hatékonyság növelése miatt tehát külön ellenőrzőösszegeket használ. Az IP fejléc hozzátétele után az eredeti üzenet így néz ki:

```
<<-------------------------  32 bit  ------------------------->>
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Verzió |  IHL  |Szolgálattípus |        Teljes hosszúság       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Azonosítás          |X|D|M|    Datagramm-eltolás    |
|                               |X|F|F|                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Élettartam  |   Protokoll   |   A fejrész ellenőrzőösszege  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Forráscím                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Célcím                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TCP fejléc, majd a tényleges adatok ...
```

Ha az IP fejlécet I-vel jelöljük, akkor az eredeti állományunk így néz ki:

```
IT....  IT....  IT....  IT....  IT....
```

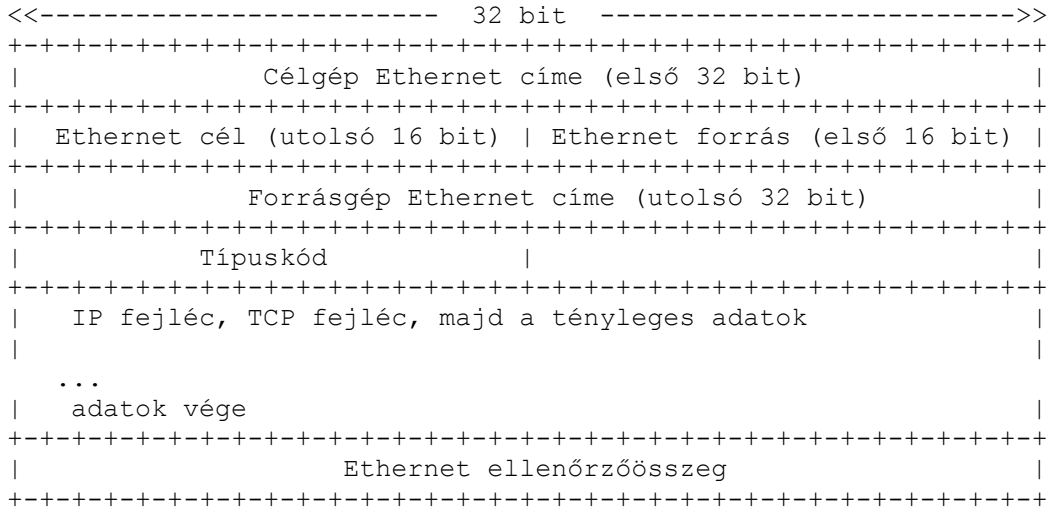
Nem esett szó a fejlécben lévő többi mező jelentéséről, mert a legtöbbjük a jelen dokumentum keretein túlmutat. A Datagramm-eltolás és a DF, MF mezők a datagrammok részeinek nyomonkövetésére használatosak. (Az XX bitet nem használják.) Egy datagrammot például akkor kell széttördelni, amikor az egy olyan hálózaton halad

keresztül, amely számára nagy falatnak mutatkozik. (Ezt részletesebben lásd később.) Az Élettartam mezőben lévő szám mindig csökken, amikor a datagramm egy rendszeren halad keresztül. Amikor eléri a nullát, a datagramm megsemmisül. Ezt az eljárást a rendszerben esetleg felépülő végtelen ciklusok miatt építették a protokollba. Persze ezek felléptének valószínűsége az ideális esetben nulla, de a jól megtervezett hálózatoknak a bekövetkezhetetlen eseményekkel is el kell tudniuk bánni. Amikor a hálózati réteg összerak egy teljes datagrammot, tudnia kell, hogy mit tegyen vele. Végül az Azonosítás mező ahhoz kell, hogy a célhoszt meg tudja állapítani, hogy egy újonnan érkezett csomag melyik datagrammhoz tartozik. Egy datagramm minden egyes darabja ugyanazzal az Azonosítás mező értékkel rendelkezik.

Lehetséges, hogy az így felépített datagrammhoz több fejléc már nem kell. Ha a küldő számítógépet a célgéphez vagy egy átjáróhoz közvetlen telefonvonal köti, akkor a datagrammokat egyszerűen kiküldi a vonalra (habár aszinkron protokoll használatakor az legalább néhány oktetet hozzátesz az elejéhez és a végéhez).

2.3 Az Ethernet szint

Manapság a legtöbb hálózat Ethernetet használ. A következőkben az Ethernet fejléccel foglalkozunk. Sajnos az Ethernetnek megvan a saját címzési módszere, mivel a létrehozók biztosítani akarták, hogy semelyik két gépnek se legyen ugyanaz az Ethernet címe. Azt is el akarták érni, hogy a felhasználónak ne kelljen a címek hozzárendelésével foglalkozni, ezért minden Ethernet vezérlő gyárilag beégetett címmel rendelkezik. Hogy ne kelljen egyetlen címet se újra kiosztani, a fejlesztők az Ethernet cím hosszát 48 bitben határozták meg. Az Ethernet vezérlőket gyártó cégeknek regisztráltatniuk kell magukat egy központnál, hogy biztosak legyenek abban: az általuk kiadott címek még nem léteznek. Az Ethernet ún. üzenetszórásos közeg, azaz olyan, mint egy partivonal. Az Ethernetre ültetett csomagot a hálózaton lévő összes gép látja, ezért valami még hiányzik, hogy azt biztosan a megfelelő gép kapja meg. Nem nehéz kitalálni, hogy itt jelenik meg az Ethernet fejléc. Minden Ethernet csomagnak egy 14 oktetes fejléce van, amely a forrás- és a célgép címét, valamint egy típuskódot tartalmaz. A hálózaton lévő gépek csak az olyan csomagokat figyelik, amelyek célmezőjében a saját Ethernet címüket találjak. (Látható, hogy milyen könnyű csalni, ezért az Ethernet hálózatok nem a legbiztonságosabbak.) Vegyük észre, hogy az Ethernet címek és az Internet címek között nincs semmiféle kapcsolat. Minden számítógépnek van egy táblázata, amelyben felsorolja, hogy milyen Ethernet cím milyen Internet címnek felel meg. (Ennek a táblázatnak a felépítését lásd később.) A címek mellett a fejlécben szerepel még egy típuskód is. Ennek segítségével ugyanazon a hálózaton többfajta protokollkészlet használata is lehetséges: TCP/IP, DECnet, Xerox, NS stb... Ezen protokollok mindegyike különböző értéket helyez a típus mezőbe. Végül ott az ellenőrzőösszeg, amelyet az Ethernet vezérlő az egész csomagra vonatkozóan számít ki. A vételkor a célgép Ethernet vezérlője is kiszámítja ezt az ellenőrzőösszeget, és ha a kettő nem egyezik, akkor eldobja a csomagot. Az ellenőrzőösszeg nem a fejlécbe, hanem a csomag végére kerül. Az üzenet tehát így néz ki:



Ha az Ethernet fejlécet E-vel, az ellenőrzőösszeget pedig C-vel jelöljük, akkor az eredeti állományunk így néz ki:

```
EIT....C  EIT....C  EIT....C  EIT....C  EIT....C
```

A csomagok megérkezésekor persze a fenti fejlécek mindegyikét leszedi a megfelelő protokoll. Az Ethernet interfész az Ethernet fejlécet és az Ethernet ellenőrzőösszeget 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. Ezzel végére is értünk a TCP/IP-be való bevezetésünknek. Mivel

vannak olyan kritikus pontok, amelyeket nem érintettünk, ezért visszamegyünk, és részletesebben tárgyalunk egyet-kettőt. (Az itt említettek részletes leírásával TCP tekintetében az RFC 793, IP tekintetében az RFC 791, illetve IP használatával Etherneten az RFC 894 és 896 dokumentumok foglalkoznak.)

3. Ismertebb socket-ek és az alkalmazási réteg

Az eddigiekben azt vettük sorra, hogy egy üzenet hogyan darabolódik szét, hogyan jut el egy géptől egy másikig, majd ott hogyan áll ismét össze. Mindez még kevés ahhoz, hogy hasznos dolgot lehessen végezni. Szükség van valamilyen módszerre, amelynek segítségével egy másik számítógéppel kapcsolatba lehet lépni, oda be lehet jelentkezni, közölni lehet vele, hogy milyen adatokra van szükségünk, illetve amellyel az adatok átvitelét szabályozni tudjuk. (Más alkalmazás, pl. elektronikus levelezés, esetén is ezzel analóg protokollra van szükség.) Ezt a feladatot az alkalmazási réteg protokolljai végzik el, amelyek a TCP/IP tetején találhatóak. Ez annyit jelent, hogy üzenet küldésekor azt a TCP-nek továbbítják, amely gondoskodik róla, hogy eljusson a célállomáshoz. Mivel a TCP és az IP kezelik a hálózati vonatkozásokat, ezért az alkalmazási protokollok a hálózatot egyszerű byte-folyamnak tekintik, mint például egy terminál- vagy telefonvonalat.

Mielőtt az alkalmazói programokkal kapcsolatban további részletekbe bocsátkoznánk, meg kell vizsgálnunk, hogy hogyan lehet egy alkalmazást megtalálni. Tegyük fel, hogy egy állományt szeretnénk küldeni a hálózaton keresztül a 124.6.4.7 IP című gépnek. A folyamat elindításához azonban az Internet címnél többre lesz szükség. A célállomás oldalán az FTP kiszolgálóval fel kell venni a kapcsolatot. A hálózati programokat általában külön feladatok elvégzésére programozzák. A különböző feladatokat (állományátvitel, bejelentkezés távoli terminálról, levelezés stb...) a legtöbb rendszerben más-más programok végzik. Amikor a fenti példában a 124.6.4.7 című géppel kapcsolatot építünk ki, akkor azt is meg kell mondanunk, hogy az ottani FTP kiszolgálóval szeretnénk kommunikálni. Ennek megvalósítására minden kiszolgáló jól ismert socket-ekkel (foglalatok, szolgálatelérési pontok) rendelkezik. Ennek magyarázataképpen tekintsük a következőket. Emlékezzünk vissza, hogy a TCP a különböző kommunikációk kézben tartására különböző portokat használ. A felhasználói programok többé-kevésbé véletlenszerűen választanak portot, de egyes portok eleve olyan programoknak felelnek meg, amelyek valamilyen kérés kiszolgálására várnak (ezek lennének a kiszolgálók). Állományátvitel esetén például egy „ftp” nevű programot indítunk el, amely a kapcsolat felépítéséhez a saját oldalán véletlenszerűen kijelöl egy portot, mondjuk az 1234-t. A céloldalon viszont a 21-es portot jelöli meg, amely az FTP kiszolgáló hivatalos portjának felel meg. Vegyük észre, hogy itt két különböző programról van szó. Az egyik az „ftp”, amelyet a küldő oldalon indítottunk el, és amely a terminálról kapott parancsokat továbbítja a másik oldalhoz; a célállomáson lévő gépen viszont az FTP kiszolgálóhoz beszélünk. Ezt arra találták ki, hogy a hálózatról parancsokat fogadjon, nem úgy mint egy interaktív terminál. Semmi szükség arra, hogy az „ftp” program jól ismert socket-et használjon. A kiszolgálókkal teljesen más az eset, hiszen a kapcsolatokban parancsokat kell tudniuk fogadni. A hivatalos portok és a hozzájuk rendelt számok az RFC 997-től kezdve az „Internet Numbers” (Internet Számok) RFC dokumentumban olvashatók (az írás pillanatában a legutóbbi az RFC 1162). Ez a dokumentum régebben az „Assigned Numbers” (Kiosztott Számok) nevet viselte.

A fentiek után nyilvánvaló, hogy egy kapcsolatot négy szám jellemez: a két Internet cím, és a két TCP port száma. Ez a négy szám minden egyes datagrammban megtalálható. (Az Internet címek az IP fejlécben, a TCP portok száma pedig a TCP fejlécben van.) Az egyediség megkövetelése miatt semelyik két kapcsolat esetén sem lehet ugyanaz mind a négy szám. Ugyanakkor elég, ha csak egy szám tér el a másik négytől. Semmi nem tiltja például azt, hogy ugyanazon a gépen lévő két különböző felhasználó állományokat vigyen át ugyanarra a távoli gépre. Ennek a megvalósítása az alábbi paraméterekkel lehetséges:

	Internet cím	TCP port száma
1. kapcsolat	128.6.4.194, 128.6.4.7	1234, 21
2. kapcsolat	128.6.4.194, 128.6.4.7	1235, 21

Mivel ugyanazokról a gépekről van szó, az Internet címek ugyanazok. Továbbá, mivel mind a két kapcsolatban állományátvitelről van szó, ezért a kapcsolatok egyik végén a jól ismert FTP port száma (21) található. Az egyetlen dolog, ami különbözik, az a felhasználók által futtatott programok portszáma. Ez tökéletesen elegendő. A kapcsolatok felépítésében az az általános gyakorlat, hogy legalább az egyik oldal utasítja a hálózati szoftvert arra, hogy számára egyedi port-ot allokáljon. A legtöbb esetben ezt a felhasználó felőli oldal teszi meg, mivel a kiszolgálónak egy mindenki által jól ismert számot kell használnia.

Most, hogy már tudjuk hogyan kell kapcsolatot felépíteni, menjünk vissza az alkalmazói programokhoz. Ahogy már fentebb említettük: miután a TCP megnyitott egy kapcsolatot, rendelkezésünkre áll egy vonal, ami akár egy egyszerű drót is lehetne. A feladat rázós részeit a TCP és az IP kezelik. Ez persze még nem elég, ugyanis tudnunk kell, hogy mit küldhetünk át a vonalon. Valójában ez nem más mint a küldhető parancsok és azok formátumának leírása. Az átküldött rész lényegében adatok és parancsok egyvelege, amiket a szövegkörnyezet különböztet meg egymástól. Például a levelezést megvalósító protokoll működése az alábbi: a felhasználói oldal levelező programja kapcsolatot épít fel a célállomás levelezést kiszolgáló programjával. A küldő program

megadja a forrásgép nevét, a küldő címét és a címzetteket. Ezek után egy parancsot küld, amelyben arról tájékoztat, hogy az üzenet szövege következik. Ettől a ponttól a kiszolgáló az adatokat nem parancsokként, hanem üzenetként értelmezi mindaddig, amíg egy speciális, az üzenet végét jelentő jellel (egy egyedül álló pont a sor elején) nem találkozik. Ez után a két oldal ismét parancsokkal kommunikál. Ez a legegyszerűbb módja az üzenetek küldésének, és a legtöbb alkalmazás így is működik.

Az állományok átvitele ennél valamivel bonyolultabb. Átvitel esetén két különböző kapcsolat épül fel. Az elején minden úgy megy mint a levelezéskor. A felhasználó programja olyan parancsokat küld, mint „jelentkeztess be ilyen és ilyen felhasználóként", „ez a jelszóm", „küldd el ezt és ezt az állományt". Miután az adatkérésre a parancs elment, a tényleges adatok átvitelére egy második kapcsolat épül fel. Persze ezt meg lehetne oldani ugyanazon az egy kapcsolaton keresztül is, ahogy a levelezés teszi. Az ok, hogy ez mégsem így történik, abban rejlik, hogy az állományátvitel általában hosszú ideig tarthat. A tervezéskor úgy érezték, hogy jobb a felhasználónak meghagyni a menet közbeni parancskiadás lehetőségét (például megszakításhoz stb... Lehetséges az is, hogy két különböző géphez nyissunk meg kapcsolatot, és egy állományt az egyiktől a másikhoz küldjünk. Ebben az esetben az adatok nem keveredhetnek a parancsokkal.)

A távoli terminálhívások egy harmadik módszert használnak. A távoli bejelentkezéskor csupán egy kapcsolat épül fel. Normális esetben ezen csak adatok mennek keresztül. Amennyiben parancsot akarunk kiadni (pl. A terminál típusának a beállítására, vagy valamilyen üzemmód átállítására), akkor egy speciális karaktert kell küldeni, amely jelzi, hogy a következő karakter parancs. Ha ezt a speciális karaktert adatként akarjuk küldeni, akkor kettőt kell egymás után küldeni.

Ebben az ismertetőben az alkalmazási protokollokról részletesen nem szólunk. Ehelyett javasoljuk a megfelelő RFC dokumentumok tanulmányozását. Az alkalmazások viszont egy sor olyan konvención alapulnak, amelyeket érdemes részletesen érinteni. Az első ilyen a közös hálózati reprezentáció: a TCP/IP-t úgy tervezték, hogy minden számítógépen alkalmazható legyen. Sajnos nem minden számítógép tárolja ugyanúgy az adatokat. A karakterek (ASCII vs. EBCDIC), a sorvég-jelek kódolásában (kocsi-vissza, soremelés), és abban, hogy a terminálok a karaktereket egyenként vagy soronként küldjék, mind különbségek mutatkoznak. A különbözőképpen működő számítógépek kommunikációjának elősegítése miatt minden egyes alkalmazói protokoll szabványos reprezentációkat definiál. A TCP és az IP azonban nem törődik a reprezentációval: a TCP egyszerűen csak okteteket küld. Az oktetek értelmezését persze mind a két oldalnak ugyanúgy kell végeznie. Az alkalmazásokat leíró RFC dokumentumok minden egyes esetben az adott alkalmazás szabványos reprezentációját definiálják. Ez a legtöbbször „tiszta ASCII" formátumnak felel meg: ASCII karakterek használata, sorvég-jelként kocsi-vissza utáni soremeléssel. A távoli bejelentkezés definiál egy „szabványos terminált" is, amely egy visszhangos, félduplex működésű terminál. A legtöbb alkalmazásban azonban arra is lehetőség van, hogy két számítógép számukra megfelelőbb reprezentációban állapodjon meg. Például a PDP-10 számítógépekben egy szó 36 bites. Két ilyen gép között lehetséges 36 bites bináris állományok átvitele. Hasonlóan, ha két rendszer inkább teljes duplex kommunikációt preferál, akkor megegyezhetnek annak használatában. Azonban mindezektől függetlenül minden egyes alkalmazásnak van egy szabványos reprezentációja, amelyet minden gépnek támogatnia kell.

3.1 Egy példa az alkalmazásokra: SMTP

Az alkalmazói protokollok szerkezetének jobb átlátása végett álljon itt az SMTP (Simple Mail Transfer Protocol—egyszerű levéltovábbítási protokoll), azaz a levelezést megvalósító protokoll egy példája. Tegyük fel, hogy a TOPAZ.RUTGERS.EDU nevű számítógép szeretné az alábbi üzenetet elküldeni.

```
Date: Sat, 27 Jun 87 13:26:31 EDT
From: hedrick@topaz.rutgers.edu
To: levy@red.rutgers.edu
Subject: meeting

Let's get together Monday at 1pm.
```

Az üzenet formátumát egy Internet szabvány (RFC 822) írja le. A szabványban megfogalmazódik, hogy az üzenetet ASCII karakterekként kell továbbítani. Az üzenet szerkezetének az alábbiak szerint kell kinéznie: fejléc sorok, aztán egy üres sor, majd az üzenet szövege következik. Végül a fejléc sorok szintaxisát definiálja részletesen: általában egy kulcsszó, majd egy érték.

A fenti üzenet címzettje LEVY@RED.RUTGERS.EDU. Kezdetben ez úgy nézett ki, hogy csak a címzett nevét és a gépet írták bele: „személy és gép". A szabványok fejlődése azonban ezt sokkal rugalmasabbá tette. Ma már más rendszerek üzeneteinek a kezelésére is vannak előírások (ami persze „magától értetődő"). Ezzel lehetővé válik az Internetbe be nem kapcsolt gépek miatti automatikus átirányítás (forwarding): például az üzenetek egy sor rendszer számára egy központi (mail server) géphez kerülnek. Egyáltalán nem szükséges tehát, hogy létezzen a RED.RUTGERS.EDU névvel jelölt számítógép. A névkiszolgálókat úgy is be lehet állítani, hogy az üzenetek

címzettet jelentő mezőjébe tanszékeket írunk, és minden egyes tanszék üzeneteit egy megfelelő számítógéphez irányítjuk. Az is lehetséges, hogy a @ jel előtti részbe ne egy felhasználónak a nevét írjuk, hanem valami mást. Egyes programokat fel lehet készíteni az üzenetek feldolgozására. A levelezési listák, illetve az olyan általános nevek, mint „postmaster" vagy „operator" kezelésére is felkészült a rendszer.

Az üzenet küldésének módját az RFC 821 és 974 dokumentumok tárgyalják. A küldést végző program párszor lekérdezi a névkiszolgálót, hogy meghatározza a célállomást. Az első lekérdezés alkalmával arról tájékozódik, hogy mely gépek kezelik a RED.RUTGERS.EDU gépnek szóló leveleket. Ebben az esetben a kiszolgáló válasza, hogy a RED.RUTGERS.EDU saját maga kezeli az üzeneteit. Ez után a program a RED.RUTGERS.EDU címét kéri le, ami 128.6.4.2. Ezek után a levelező program egy TCP kapcsolatot nyit meg a 128.6.4.2 gép 25-ös portjára. A 25-ös port a leveleket fogadó foglalatnak felel meg. Miután a kapcsolat létrejött, a levelező program megkezdi a parancsok küldését. Az alábbiakban álljon itt egy tipikus kommunikáció. A sorok előtt az szerepel, hogy az a TOPAZ vagy a RED nevű géptől származik-e. A példában TOPAZ kezdeményezte a kapcsolatot:

```
RED     220 RED.RUTGERS.EDU SMTP Service at 29 Jun 87 05:17:18 EDT
TOPAZ   HELO topaz.rutgers.edu
RED     250 RED.RUTGERS.EDU - Hello, TOPAZ.RUTGERS.EDU
TOPAZ   MAIL From: <hedrick@topaz.rutgers.edu>
RED     250 MAIL accepted
TOPAZ   RCPT To: <levy@red.rutgers.edu>
RED     250 Recipient accepted
TOPAZ   DATA
RED     354 Start mail input; end with <CRLF>.<CRLF>
TOPAZ   Date: Sat, 27 Jun 87 13:26:31 EDT
TOPAZ   From: hedrick@topaz.rutgers.edu
TOPAZ   To: levy@red.rutgers.edu
TOPAZ   Subject: meeting
TOPAZ
TOPAZ   Let's get together Monday at 1pm.
TOPAZ   .
RED     250 OK
TOPAZ   QUIT
RED     221 RED.RUTGERS.EDU Service closing transmission channel
```

A parancsokban mindenütt normál szöveg szerepel: ez az Internet szabványokra tipikusan jellemző. A protokollok többsége szabványos ASCII parancsokat használ, ami arra is jó, hogy követhessük éppen mi történik, és a problémákat diagnosztizálni lehessen. A levelező program például minden ilyen beszélgetést egy állományban naplóz. Ha valami nem a megfelelő módon történik, akkor az állományt elküldhetjük a postmaster-nak. Mivel ez ASCII fomátumú, ezért látni, hogy mi történt. A dolog arra is jó, hogy közvetlenül a levelezést kiszolgáló géppel lépjünk kapcsolatba tesztelés céljából. (Néhány újabb protokoll annyira összetett, hogy ez nem praktikus. A parancsoknak olyan szintaxis felelne meg, amely szintaktikus elemzőt igényelne. Ez azt jelenti, hogy az újabb protokoll esetében a tendencia a bináris formátumok felé mutat. Általában olyan struktúrákról van szó, mint a C vagy a Pascal nyelvek rekordja.) Második észrevételként említjük, hogy a válaszok mindegyike számmal kezdődik: ez is az Internet protokollok jellemző vonása. A megengedett válaszokat a protokollok definiálják. A számok segítségével a felhasználói programok egyértelműen kommunikálhatnak. A válaszok maradék része szöveg, ami a könnyebb olvashatóság miatt szerepel, és nincs semmiféle kihatása a programok működésére. (Habár van egy pont, ahol a protokoll a válasz szövegének egy részét is felhasználja.) Maguk a parancsok arra használatosak, hogy a levelező program a kiszolgálóval közölje azokat az információkat, amelyek az üzenet továbbítása miatt szükségesek. A fenti kiszolgáló az információt az üzenetből is kiolvashatja. Bonyolultabb esetekben azonban ez nem lenne biztonságos. Minden kommunikáció a HELO paranccsal kezdődik, amit a kapcsolatot kezdeményező rendszer nevének kell követnie. Ezek után kovetkezik a küldő és a címzett meghatározása. (Lehet több RCPT parancsot is kiadni, ha több címzett van.) Végül maga az üzenet jön. A szöveget olyan sorral fejezzük be, amiben csak egy pont szerepel. (Ha a szövegben is szerepel ilyen sor, akkor a pont megduplázódik.) Miután az üzenet fogadása megörtént, a küldő másik üzenetet küldhet, vagy befejezheti a kommunikációt, mint ahogy a fenti példában is történt.

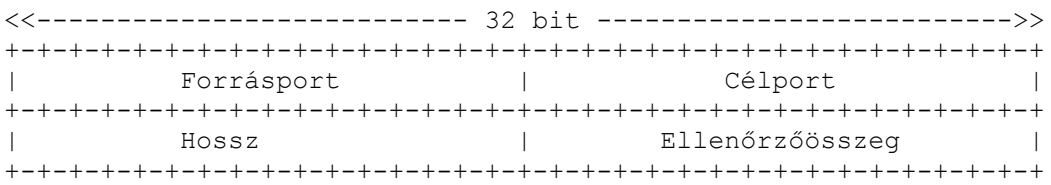
A válaszokat jelölő számokat egy minta szerint képezzük. A protokoll definiálja azokat a válaszokat, amelyek egy adott parancsra adhatóak. Amennyiben nem érdekes a válaszok részletes elemzése, akkor elég csak az első számjegyet figyelembe venni. A 2-vel kezdődő válaszok sikeres parancsot jelölnek. A 3-mal kezdődőek esetén további parancsokat vár a kiszolgáló (ld. fent is). A 4 ideiglenes hibát jelez (pl. Lemezterület megtelt). Az üzenetet ilyenkor el kell menteni, és később újra kell próbálkozni. Az 5 állandó jellegű hibára utal, például nem létezik a címzett. Az üzenetet egy hibaüzenet kíséretében vissza kell küldeni a feladónak.

(A fejezetben említett protokollokkal kapcsolatban további információt szolgáltat az RFC 821/822 a levelezésről, az RFC 959 az állományátvitelről, és az RFC 854/855 a távoli bejelentkezésről.)

4. Nem TCP protokollok: UDP és ICMP

Eddig csak olyan kapcsolatokkal foglalkoztunk, amelyek TCP-t használnak. Emlékezzünk vissza, hogy a TCP az üzenetek datagrammokra darabolásáért és helyes sorrendben történő visszaállításáért felelős. Sok alkalmazás során találjuk magunkat szembe olyan üzenetekkel, amelyek elférnek egyetlen datagrammban is. Egy példa erre a nevek kikeresése. Amikor egy felhasználó egy másik rendszerrel kapcsolatba akar lépni, akkor általában az adott rendszer nevét fogja megadni, és nem az IP címét. Mielőtt bármit is kezdhetne vele, a felhasználó rendszerének ezt a nevet le kell fordítania IP címre. Az erre a célra szolgáló adatbázissal viszont nem minden rendszer rendelkezik, ezért a felhasználó rendszere az adatbázissal bírót kéri meg a fordításra. A kérés annyira rövid, hogy biztosan elfér egyetlen datagrammban. Ugyanez mondható el a válaszról is. Úgy látszik, hogy nem érdemes a TCP-t használni. Persze a TCP az üzenetek darabolásán kívül még mást is csinál. Biztosítja, hogy az üzenetek megérkezzenek: ahol szükséges, ott a datagrammokat újraadja. Viszont az olyan kérdéshez, amely egyetlen datagrammban elfér, nincs szükség a TCP teljes bonyolultságára. Ha egy pár másodpercen belül nem kapunk választ, akkor egyszerűen megismételjük a kérdést. Az ilyen alkalmazásokra a TCP mellett létezik más alternatíva.

A legszélesebb körben használt ilyen protokoll az UDP (user datagram protocol—felhasználói datagrammprotokoll), amelyet olyan alkalmazásokhoz találtak ki, ahol nincs szükség datagramok sorozatba állítására. Hasonlóképpen illeszkedik a rendszerbe, mint a TCP. A hálózati szoftver az adatok elejére ráilleszti az UDP fejlécet ugyanúgy, ahogy a TCP fejléc esetében teszi. Az UDP ezek után az IP-nek továbbítja az adatot. Az IP hozzáteszi a saját fejlécét, amibe a TCP helyett az UDP protokollszámát helyezi el a Protokoll mezőben (lásd IP fejléc). Az UDP nem végez annyi feladatot, mint a TCP: nem tördeli szét az üzenetet datagrammokra, nem figyeli a már elküldött adatokat, hogy majd esetleg újraadja őket. Az UDP csak portszámokat biztosít, hogy egyszerre több program is használhassa a protokollt. Az UDP portszámok ugyanúgy használatosak, mint a TCP portszámok. Az UDP-t használó kiszolgálókhoz is léteznek jól ismert portszámok. Megjegyezzük még, hogy az UDP fejléc sokkal rövidebb, mint a TCP fejléce. Ebben is szerepel a forrás- és a célport száma, valamint egy ellenőrző összeg, de ennyi az egész. Nincs benne sorszám, mert nincs szükség rá. Az UDP fejléc így néz ki:



Az UDP-t a nevek kikeresését végző (lásd az IEN 116, az RFC 882 és az RFC 883 dokumentumokat), illetve az ezekhez hasonlóan működő protokollok használják.

Egy másik alternatív protokoll az ICMP (Internet control message protocol) internet vezérlőüzenet protokoll) nevet viseli. Az ICMP-t a hibaüzenetek és a TCP/IP-t megvalósító szoftvernek szánt üzenetek kezelésére használják. Kapcsolat kérésekor a kezdeményező rendszer kaphat például olyan ICMP üzenetet, hogy „host unreachable” (elérhetetlen gép). Az ICMP-t használják még arra is, hogy magáról a hálózatról információkat gyűjtsenek. A protokollt az RFC 792 dokumentum írja le teljes részletességében. Az ICMP abban hasonlít az UDP-hez, hogy mindketten olyan üzenetekkel foglalkoznak, amelyek egyetlen datagrammban elférnek. Az ICMP azonban annál is egyszerűbb. Még csak portszámok sincsenek a fejlécében. Mivel minden egyes ICMP üzenetet maga a hálózati szoftver értelmez, ezért nincs szükség olyan portszámokra, amelyek megmondják, hogy egy adott ICMP üzenet hova menjen.

5. Név- és információszervezés: a tartomány (domain) rendszer

Ahogyan már korábban jeleztük, a hálózati szoftvernek egy 32 bites Internet címre van szüksége ahhoz, hogy egy kapcsolatot felépíthessen, vagy hogy datagrammokat küldhessen. A felhasználók viszont inkább a számítógépek neveivel mintsem számokkal szeretnének hivatkozni rájuk (a neveket könnyebben meg lehet jegyezni). Ezért létezik egy adatbázis, amelyből a hálózati szoftver kikeresheti a névnek megfelelő címet, és fordítva. Amikor az Internet még nem volt ilyen kiterjedt, akkor ez viszonylag könnyen megoldódott: minden gépnek volt egy adatállománya, amelyben az összes többi rendszer nevét és címét felsorolták. Ma már túl sok rendszer létezik ahhoz, hogy az ilyen megközelítés praktikus legyen. Emiatt ezeket az állományokat olyan névkiszolgálók váltották fel, amelyek a gépek neveit és a megfelelő címeket tartják nyilván. (A sokfajta információ közül ez csak egy. Valójában ezek a kiszolgálók sokkal általánosabb feladatot látnak el.) A valóságban egyetlen központi gép helyett az ilyen kiszolgálók egymással összekapcsolt halmaza használatos. Manapság már olyan sok különböző intézmény kapcsolódik az Internethez, hogy nem lenne praktikus, ha egy

központi hatóságot kellene értesíteniük minden olyan esetben, amikor egy gépet a hálózatba be- vagy abból kikapcsolnak. Éppen ezért a névadásra az egyes intézmények a rendszerükön belül saját maguk jogosultak. Az így kialakított névkiszolgálók közösen egy fa struktúrát alkotnak, amely az intézmények hálózati szerkezetének felel meg. Ezt a szerkezetet a nevek is tükrözik. Tipikus példa erre a BORAX.LCS.MIT.EDU név, amely a MIT számítástechnikai laboratóriumának (LCS) egy számítógépét jelöli (ilyen példa lehetne még: maxi.inf.elte.hu, ami az ELTE Általános Számítástudományi tanszékének maxi nevű gépét adja). A gép Internet címének meghatározásához 4 potenciális kiszolgálót kellene megkérdezni. Először egy központi kiszolgálótól (root - gyökér, ld. a fa struktúrát) kellene megtudakolni, hogy hol található az EDU kiszolgáló, amely nem más, mint a hálózatba kapcsolt oktatási intézmények nyilvántartása. A gyökérként szereplő kiszolgáló több EDU kiszolgáló nevét és Internet címét adná meg. (Minden szinten több ilyen névkiszolgáló van, hogy az esetleges meghibásodások ne okozzanak fennakadást.) A következő feladat lenne az EDU kiszolgáló lekérdezése a MIT névkiszolgálójáról. Itt is több kiszolgáló nevét és Internet címét kapnánk meg. Ezek közül általában nem mindegyik található az intézmény területén (egy esetleges áramszünet fellépte miatt). Ez után a MIT-től kérdeznénk le a számítástechnikai laboratórium (LCS) névkiszolgálójának adatait, majd végül a laboratóriumi névkiszolgálók egyike adná a BORAX adatait. A végső eredmény a BORAX.LCS.MIT.EDU gép Internet címe lenne. A fenti szintek mindegyike egy tartományt (domain) jelöl. A teljes BORAX.LCS.MIT.EDU név pedig egy tartománynév (domain name). (Ugyanígy a felsőbb tartományok nevei is tartománynevek: LCS.MIT.EDU, MIT.EDU és EDU.)

Az esetek nagy többségében szerencsére nem kell a fenti lépések mindegyikét végrehajtani. A legfelső kiszolgáló (gyökér) ugyanis egyben a legfelső szinten lévő tartományok (pl. EDU) névkiszolgálójaként is szerepel. Tehát a gyökér kiszolgáló felé irányuló egyetlen kérdéssel a MIT névkiszolgálójához lehet eljutni. Az alkalmazott szoftverek pedig a már feltett kérdésekre kapott válaszokra emlékeznek. Ez azt jelenti, hogy a LCS.MIT.EDU kiszolgáló lekérdezése után tudja, hogy hol keresse a LCS.MIT.EDU, a MIT.EDU és az EDU tartománybeli kiszolgálókat. A BORAX.LCS.MIT.EDU fordítására szintén emlékszik. Persze minden ilyen információnak van egy megfelelő élettartama, ami tipikusan pár napnak felel meg. Az élettartam lejárta után az információkat fel kell frissíteni. Az intézmények ilyen módon változtathatnak, ha akarnak.

A tartományrendszer feladata nem merül ki az Internet címek megtalálásában. Minden egyes tartománynév csomópontként szerepel egy adatbázisban. A csomópontnak különböző tulajdonságokat jellemző rekordjai lehetnek. Ilyen az Internet cím, a számítógép típusa, és a számítógép által biztosított szolgáltatások felsorolása. Egy program egy adott névvel kapcsolatban kérheti ezen információk valamelyikét, vagy az összeset. Megoldható az is, hogy egy adatbázisbeli csomópont egy másik csomópont álneveként (alias) szerepeljen. Az is lehetséges, hogy a tartományrendszerben felhasználókról, levelezési listákról, vagy más objektumokról tároljunk adatokat.

A fenti adatbázisok működését, illetve az azok lekérdezését megvalósító protokollokat is Internet szabvány írja le. Minden hálózati alkalmazásnak meg kell tudnia valósítani ezeket a lekérdezéseket, mivel hivatalosan így történik a hosztnevek kiértékelése. Az alkalmazások általában saját rendszerükön (tartományukon) belül keresnek egy névkiszolgálót. Ez a kiszolgáló aztán a felsőbb szinten (az ő tartományán) lévő kiszolgálókkal veszi fel a kapcsolatot. Ezzel a módszerrel az alkalmazásokban lévő kód mennyiségét lehet lecsökkenteni.

A tartományrendszer fontos szerepet tölt be az elektronikus levelezésben. Az adatbázisokban szerepelhetnek olyan bejegyzések, amelyek megmondják, hogy melyik gép kezeli egy adott név leveleit, egy felhasználó levelei hová érkezzenek, illetve levelezési listákat is definiálhatnak.

(A tartományrendszerről az RFC 1034, 1035 és 1101 dokumentumok írnak. Ezek régebbi verziói: RFC 882, 883 és 973. Az RFC 974 pedig a tartományrendszernek az elektronikus levelezésben betöltött szerepéről szól.)

6. Útvonal-választás

A fentiekben említettük, hogy az IP implementációknak gondoskodniuk kell a datagramm célcím által jelzett címre való eljuttatásáról. Azt azonban nem írtuk le, hogy ez hogyan is történik. Egy datagramm rendeltetési helyére juttatásának mikéntjét az útvonal-választás (routing) kifejezés jelöli. A részletek nagymértékben függenek az adott implementációtól, viszont egy-két dolgot általánosságban el lehet mondani.

Először is az szükséges, hogy az IP-t megvalósító modellel tisztában legyünk. Az IP alapállapotban azzal a feltevéssel él, hogy a rendszerek valamilyen lokális hálózatra kapcsolódnak. Feltesszük, hogy a rendszer a saját hálózatán keresztül datagrammokat tud küldeni egy másik rendszernek. (Ethernet alapú hálózat esetén egyszerűen a célállomás Ethernet címét kell megkeresnie, majd a datagrammot ki kell adnia a hálózatra.) A probléma akkor jelentkezik, amikor egy másik hálózaton lévő rendszerhez kell küldeni datagrammot. Itt lépnek be az átjárók (gateway). Az átjáró egy olyan hálózati eszköz, amely egy hálózatot két vagy több másikkal köt össze. Ez a gyakorlatban legtöbbször egy olyan számítógépet jelent, amelynek több hálózati interfésze van. A Rugers University-nél például van egy Unix alapú gép, amelynek két különböző Ethernet interfésze van. Ťgy az kapcsolódik a 128.6.4, és a 128.6.3 hálózathoz. Ez a számítógép a két hálózat között átjáróként üzemelhet. A

hálózati szoftvert úgy kell beállítani, hogy az átjáró a két hálózat között datagrammokat tudjon küldeni. Ha egy gép a 128.6.4 hálózatról olyan datagrammot küld az átjáró felé, amely a 128.6.3 hálózaton lévő gépek egyikének szól, akkor azt az átjáró továbbítja a célállomás felé. A főbb kommunikációs központokban több átjáró is található, amelyek különböző hálózatokat kötnek össze egymással. (A legtöbbször speciálisan erre a feladatra készített átjárókat alkalmaznak, amelyek megbízhatóbban, és sokkal hatásosabban működnek az általános célú átjáróknál. Sok cég kínál ilyen rendszereket.)

Az IP szerinti útvonal-választás teljes mértékben a célállomás hálózati számán alapszik. A hálózatba kötött minden egyes számítógép rendelkezik egy táblázattal, amelyben a hálózati számokat tárolják. Minden hálózatszámhoz tartozik egy átjáró, amelyen keresztül az adott hálózathoz eljuthatunk. Azt észre kell venni, hogy az átjáró nincs feltétlenül arra a hálózatra kötve: egyszerűen csak az a legjobb út, amelyen keresztül az adott hálózathoz el lehet jutni. Például a Rutgers Egyetem NSFnet kapcsolata a John von Neumann Supercomputer Center-en (JvNC) keresztül valósul meg. A JvNC-vel való összeköttetést egy nagysebességű soros vonali kapcsolat adja, amely a 128.6.3.12 című átjáróhoz van kötve. A 128.6.3 hálózaton lévő rendszerek a legtöbb egyetemen kívüli hálózat felé a 128.6.3.12 átjárót fogják használni. A 128.6.4 hálózaton lévő rendszerek viszont a 128.6.4.1 átjárót használják ugyanazon hálózatok felé. A 128.6.4.1 a 128.6.4 és a 128.6.3 hálózatok között működik átjáróként, tehát a JvNC-vel első lépésként ezen keresztül lehet kapcsolatba lépni (a 128.6.4 hálózatról).

Amikor egy számítógép datagrammot akar küldeni egy másiknak, akkor először azt ellenőrzi, hogy a fogadó nincs-e a saját hálózatán. Ha ott van, akkor a datagrammot közvetlenül neki küldi el. Ha nincs ott, akkor a rendszer keresni kezdi a táblázatban a célállomás hálózati számát, és a datagrammot annak a hálózatnak az átjárója felé küldi. A hálózati számokat és átjárókat felsoroló táblázat esetenként igen nagy terjedelemre tehet szert. Az Internet például több száz hálózatot foglal magába. Különböző stratégiákat dolgoztak ki annak érdekében, hogy az útvonal-választási táblák méretét a lehető legkissebb értéken tartsák. Az egyik ilyen módszer az alapértelmezett útvonalak használata. Gyakran fellép az az eset, hogy egy hálózatból csak egyetlen átjárón keresztül lehet kijutni. Egy ilyen átjáró például egy Ethernet alapú lokális hálózat és egy gerinchálózat között létesíthet kapcsolatot. Ilyenkor persze nincs szükség arra, hogy az útvonal-választási táblában az összes külső hálózat szerepeljen. Az átjárót egyszerűen alapértelmezettnek definiáljuk, és így a választott útvonallal nem rendelkező datagrammok egyenesen az átjáróhoz kerülnek. Egy így beállított átjáró akkor is használható, ha egy hálózaton több is működik belőle. Az átjárókat úgy tervezték, hogy a „Nem ez a legjobb átjáró -- használd inkább ezt és ezt.” üzenetet generálni tudják. (Az üzenetet az ICMP-n keresztül adják le. Lásd az RFC 792-t.) A hálózati szoftverek többsége ezt az üzenetet használja arra, hogy az útvonal-választási táblájába bejegyzéseket helyezzen el. Tegyük fel, hogy a 128.6.4 hálózatnak két átjárója van: a 128.6.4.1 és a 128.6.4.59. Az első a Rutgers belső hálózataival, a második pedig közvetetten az NSFnet-tel tart kapcsolatot. Tegyük fel továbbá, hogy alapértelmezett átjáróként a 128.6.4.59-t állítottuk be, és az útvonal-választási táblában nincs más bejegyzés. Mi történik, ha a MIT hálózatára akarunk datagrammot küldeni? A MIT hálózati száma 18. Mivel ilyen bejegyzés nincs a táblázatban, ezért a datagramm egyenesen a beállított géphez, a 128.6.4.59-hez kerül. Ez persze a rossz átjáró. A datagrammunkat a 128.6.4.1-hez fogja továbbítani. Ezen kívül egy hibaüzenetet is küld nekünk „a 18-as hálózathoz használd a 128.6.4.1 átjárót” szöveggel. A hálózati szoftverünk pedig bejegyzi az adatot a táblázatba. Ennek eredményeképpen a MIT felé irányuló jövőbeli datagrammok egyenesen a 128.6.4.1 átjáró felé mennek. (A hibaüzenet küldéséhez az ICMP használatos. Ezt a fajta üzenetet ICMP átirányításnak (ICMP redirect) hívják.)

Az IP szakértők többsége azon a véleményen van, hogy a hálózati számítógépek ne próbálják meg az egész hálózat forgalmát nyomon követni. Ehelyett azt ajánlják, hogy alapértelmezett átjárókat használjanak, és rájuk támaszkodjanak az útvonalak megállapításánál, ahogy azt a fentiekben is leírtuk. Arról nem volt szó, hogy az átjárók hogyan határozzák meg az útvonalakat. Az esetükben a fenti stratégia nem használható, hiszen az útvonal-választási táblázatuknak megfelelően teljesnek kell lennie. Ezért valamiféle útvonal-választási protokoll jelenléte szükséges, amely azt írja le, hogy az átjárók hogyan találhatják meg egymást, és hogyan frissíthetik az adatbázisukat a különböző hálózatokhoz vezető (legjobb) útvonalakról. Az átjárók tervezéséről és az útvonalak megválasztásáról az RFC 1009 ad áttekintést. Az útvonal-választás legutóbbi leírását az RFC 1716 és RFC 1812 adja. A rip.doc (RFC 1058) dokumentum valószínűleg jobb bevezetést nyújt a témába. Tartalmazza egy kissebb bevezető oktatás anyagát, valamint a leggyakrabban használt útvonal-választási protokoll részletes leírását is.

7. Bővebben az Internet címekről: alhálózatok és üzenetszórás

Az eddigiekből már kiderült, hogy az Internet címek 32 bites számok, amelyeket négy (decimális) oktet formájában írunk le, pl.: 128.6.4.7. Ténylegesen háromfajta cím létezik. A probléma abban gyökeredzik, hogy a címnek a hálózatot is, és a hálózati gépet is jelölnie kell. Kezdetben úgy tartották, hogy rengeteg hálózat fog létrejönni. Ezek közül sok lesz majd kisméretű, de valószínűleg 24 bit kell az IP címek leírására. Azt is felvetették, hogy néhány nagyméretű hálózatnak 24 bit kell majd a gépei IP címének a leírására. Úgy látszott,

hogy 48 bites címeket kell bevezetni. A tervezők azonban 32 bites címeket akartak használni. A megoldás a kettő között fekszik. Feltételezték, hogy a legtöbb hálózat kisméretű lesz. Háromfajta címtartományt vettek fel. Az 1 és 126 közötti számokkal kezdődő címek a négyből csak az első oktetet használják a hálózat megcímzésére. A maradék három oktet, azaz 24 bit, jelölheti a gépeket. Az így konstruált címeket nagyméretű hálózatok használják. A címzés ezekből viszont csak 126 darabot enged meg. Ilyen hálózat az Arpanet és még egy pár nagy kereskedelmi hálózat. A csatlakozó intézmények közül kevesen kapnak ilyen „A osztályú" IP címet. A hálózaton a leggyakoribb a „B osztályú" cím, amikor a négy oktetből az első kettő a hálózat (128.1-től 191.254-ig), a maradék kettő (tehát 16 bit) pedig a gépek megcímzésére szolgál. (A hálózat címében a 0 és a 255 nem használható a lejjebb olvashatóak miatt. A 127 szintén tiltott, mert ez speciális célokra van fenntartva.) Az így kialakított címzés egy hálózaton belül tehát 64516 gépet engedélyez. (Lehetőség van több B osztályú cím felvételére is, ha ez kevés lenne.) Végül pedig a „C osztályú" címek azok, amelyekben az első három oktet jelöli a hálózatot (192.1.1-től 223.254.254-ig). Az ilyen hálózatokhoz maximum csak 254 gép csatlakozhat, de ezekből sok ilyen lehetséges. A 223-nál nagyobb számokkal kezdődő címeket „D" és „E" osztályként jövőbeli használatra tartalékolják. (A D osztályú címeket úgynevezett csoportcímzésre (multicasting) használják; ezek címzése 224.0.0.0-tól 239.255.255.255-ig tart.)

Sok hálózat számára hasznos, ha a hálózati címét felosztja alhálózatokra. A Rutgers egyetem például a B osztályú 128.6 címen érhető el. A harmadik oktetet arra használja, hogy a helyi Ehternet alapú hálózatokat megkülönböztesse egymástól. Ennek a felosztásnak ez egyetemen kívül semmilyen jelentősége nincs. A többi intézmény ebből semmit sem vesz észre. A címzéskor nem nézik a harmadik oktetet. A Rutgers-en kívüli gépek továbbra is ugyanazon az úton fogják küldeni a datagrammokat mind a 128.6.4, mind a 128.6.5 hálózatra. Az egyetemen belül azonban ez nem igaz. Minden egyes egyetemi átjárónak külön bejegyzése van az egyetemen található összes alhálózatról, míg az egyetemen kívüli átjáróknak csak a 128.6-ról van bejegyzésük. A fenti elosztást úgy is meg lehetett volna valósítani, ha az egyetem az alhálózataira C osztályú címeket kapott volna. Ezzel persze az egyetemen kívüli világ számára lett volna komplikáltabb a dolog, hiszen minden átjárónak az összes ilyen címet be kellett volna jegyeznie a táblázatába. Ez pedig az útvonalak nyomonkövetését nehezebbé tette volna. A B osztályú cím felosztásával mintegy elrejthető a belső felépítés, és így sok vesződségtől kímélünk meg másokat. Az alhálózatok ilyen felosztása speciális igényeket támaszt a hálózati szoftver felé.

Az IP címekben a 0 és a 255 speciális jelentéssel bír. A 0 az olyan gépek számára van fenntartva, amelyek nem tudják a hálózati címüket. Bizonyos helyzetekben lehetséges, hogy egy számítógép nem tudja, melyik hálózatra csatlakoztatták. A 0.0.0.23 például egy olyan számítógép címe, amelynek hosztszáma 23, de nem tudni, hogy melyik hálózaton.

A 255-t üzenetszórásra (broadcast) használják. Az üzenetszórás lényegében egy olyan üzenet, amelyet az adott hálózaton minden számítógép lát. Olyankor használatos, amikor a „címzett ismeretlen". Tegyük fel például, hogy egy, a hálózatra kapcsolt számítógép nevére van szükségünk, mert az Internet címét szeretnénk tudni. Mondjuk, hogy a legközelebbi névszolgáltatónak nem tudjuk a címét. Ilyenkor segít az üzenetszórás. Előfordulhat az is, hogy egy információt több rendszerrel meg szeretnénk osztani. Ilyenkor hatásosabb az üzenetszórás, mintha az érdekelt rendszerekhez külön küldenénk datagrammokat. Az üzenetszórás megvalósításához egy olyan IP címet kell formálni, amelyben a hálózatot jelölő részbe a küldő hálózat címét, a gépet jelölő részbe pedig csupa egyes bitet (azaz 255-t) írunk. A 128.6.4 hálózaton ez így nézne ki: 128.6.4.255. Az üzenetszórás tényleges megvalósítása az adott közegtől függ. Az Arpanet-en és két gép közötti hálózatokon nem lehet üzenetszórást alkalmazni, ellentétben az Ethernet alapú hálózatokkal, ahol a csupa egyes bitből álló Ethernet című üzenetet az azon a hálózaton lévő minden számítógép veszi.

Annak ellenére, hogy a 128.6.4 hálózaton az üzenetszórás hivatalos alakja 128.6.4.255, egyes megvalósításokban létezik erre más cím is. A szabvány megengedi a 255.255.255.255 használatát is, amely az adott lokális hálózat összes gépének szóló üzenetet jelenti. Sokszor egyszerűbb ezt a címet használni, mint a lokális hálózat címével a fenti módon megformálni az üzenetet. Ezekhez jön hozzá az a tény, hogy egyes korai megvalósításokban a 255 helyett a 0-t használták üzenetszórásra, azaz a fenti példában 128.6.4.0 lenne az üzenetszóró cím. (Nevezetesen a Berkeley Unix egyik kezdeti változatának TCP/IP kódjáról van szó. A hibát azóta természetesen kijavították, de a „félreértés" az abból származtatott egyes kereskedelmi rendszerekben tovább él -- a fordító.) Végül léteznek olyan régebbi rendszerek, amelyek egyáltalán nem ismerik az alhálózat fogalmát, számukra a fenti hálózatot a 128.6, és így az üzenetszórást a 128.6.255.255 vagy a 128.6.0.0 cím jelenti. Addig, amíg az üzenetszórás körüli kavalkád nem tisztul, igen veszélyes dologgá is válhat (szerintem ez ma már kevésbé igaz; a rendszerek 99%-a a 255-t használja—a fordító).

Mivel a 0 és a 255 speciális célokra használatos, ezért a hálózati gépek címeiben ennek a két számnak nem szabad szerepelnie. Az IP címek nem kezdődhetnek se 0-val, se 127-tel, se 223-nál nagyobb számmal. Az ezeket a szabályokat megszegő címekre Marslakókként hivatkoznak, mert elterjedt, hogy a Mars Központ Egyeteme a 225-ös hálózatot használja.

8. Datagrammok fragmentálása és összerakása

A TCP/IP-t úgy tervezték, hogy különböző hálózatokon is használható legyen. Sajnos a hálózati tervezők nem igazán értenek egyet abban, hogy maximálisan mekkora lehet egy csomag mérete. Az Ethernet hálózatoknál ez 1500 oktet. Az Arpanet maximum 1000 oktet körüli csomagokkal dolgozik. Egyes gyors hálózatoknál a csomagméret ennél jóval nagyobb lehet. Az első ötlet az, hogy az IP egyszerűen a lehető legkissebb csomagmérettel dolgozzon. Ez azonban a hatásfokot jelentősen rontaná. Nagy állományok esetén ugyanis sokkal eredményesebb a nagyobb csomagméret. Ezért a lehető legnagyobb méretet akarjuk elérni, de úgy, hogy a csak kissebb méreteket kezelő hálózatok is részt vehessenek az adatforgalomban. A következő két módszer szerint járnak el. A TCP-t úgy tervezték, hogy képes a datagramm méretet egyeztetni (negotiate). Ez azt jelenti, hogy a TCP kapcsolat felépítésekor mindkét oldal közli a másikkal az általa kezelhető maximális méretet, majd a továbbiakban a kissebbiket használják. Ťgy a nagyobb datagrammokat kezelni képes megvalósítások azokat használják, de ugyanakkor a kissebb datagrammokat ismerő implementációkkal is szót értenek. A probléma még korántsem megoldott. Ugyanis a két oldal nem feltétlenül tudja, hogy mi történik a datagrammokkal útközben. Például a Rutgers és a Berkeley egyetemek közötti adatforgalom esetén valószínű, hogy mindkettő számítógép Ethernet alapú hálózaton helyezkedik el. Ezért mindketten megértik az 1500 oktetes datagrammokat. Útközben az adatok az Arpanet-en keresztül továbbítódnak. Ez a hálózat nem tud 1500 oktetes datagrammokat kezelni, ezért azokat fragmentálnia, tördelni kell. Az IP fejléc mezői jelzik, ha a datagramm fragmentált, és az összerakásra vonakozóan is elegendő információt tartalmaznak. Ha egy átjáró egy Ethernet alapú hálózatot köt össze az Arpanet-tel, akkor annak képesnek kell lennie 1500 oktetes Ethernet csomagok fogadására és azok Arpanet méretűvé tördelésére. A TCP/IP minden megvalósításának képesnek kell lennie a darabok fogadására és az eredeti datagramm összerakására (reassembly).

A TCP/IP implementációk különböznek egymástól a datagramm méretének megválasztásában, azonban a szabvány szerint legalább 576 oktet nagyságú datagrammokat választanak, ha nem biztosak abban, hogy a nagyobb méretet útközben mindenhol megértik. Ez az eléggé konzervatív megközelítés abból fakad, hogy az összerakást megvalósító kódok sokszor hibásak. A tervezők kerülni igyekeznek a fragmentálást. Mindegyikük másként gondolkodik arról, hogy mikor biztonságos a nagyobb méret. Néhányan csak a lokális hálózatra esküsznek, de vannak olyanok is, akik az egész hálózatra kiengednek ilyen datagrammokat. Az 576 oktet eléggé biztonságos ahhoz, hogy mindenki támogassa.

9. Az Ethernet és az ARP

A korábbiakban röviden kitértünk arra, hogy az Ethernet alapú hálózatokon hogyan néz ki egy IP fejléc. Szó volt az Ethernet fejlécről és az elenőrzőösszegről is. Azt azonban nem tudtuk meg, hogy egy adott IP cím esetén milyen Ethernet címet használjunk. Erre a kérdésre egy protokoll, az ARP (address resolution protocol -- címleképezési protokoll) adja meg a választ. (Vigyázat: az ARP nem IP-beli protokoll. Az ARP datagrammok nem kapnak IP fejlécet.) Tegyük fel, hogy a 128.6.4.194 rendszerről a 128.6.4.7 rendszerrel szeretnénk kapcsolatba lépni. A kezdeményező rendszer első lépésként azt találja, hogy a 128.6.4.7 is ugyanazon az Ethernet alapú hálózaton található. Második lépésként a 128.6.4.194 megnézi, hogy szerepel-e a saját ARP táblázatában a 128.6.4.7 címen bejegyzés (a 128.6.4.7 Ethernet címe). Ha igen, akkor a datagrammhoz egy Ethernet fejlécet csatol, és elküldi. Tegyük fel azonban, hogy nincs ilyen bejegyzés az ARP táblázatban. Ťgy a csomagot nem lehet elküldeni, hiszen nincs meg az Ethernet cím. Itt jön be az ARP. A 128.6.4.169 rendszer egy „Kérem a 128.6.4.7 Ethernet címét” tartalmú ARP kérést ad ki az Ethernet hálózatra. Az adott hálózaton minden rendszer figyeli az ARP kéréseket. Ha egy rendszer olyan ARP kérést fog, amely rá vonatkozik, akkor válaszolnia kell. A fenti példában tehát a 128.6.4.7 hallja a kérést, és egy ARP üzenetet küld a 128.6.4.169-nek, amelynek tartalma: „A 128.6.4.7 Ethernet címe 8:0:20:1:56:34”. (Emlékeztetőül: az Ethernet címek 48 bitesek. Ez 6 oktetet jelent. Megegyezés szerint hexadecimális alakban, a fenti központozással írjuk a címeket.) A kérést adó rendszer a kapott információt bejegyzi az ARP táblázatába. Az esetek nagy részében az ARP táblázatokat gyorsítótárként (cache) használják: a régóta nem használt bejegyzéseket kitörlik.

A fentiekből valószínű kiderült, hogy az ARP kéréseket üzenetszórás formájában kell a hálózatra kiadni. Nem lehet azokat közvetlenül a keresett rendszerhez küldeni, hiszen a lényeg éppen a cím keresése. A kérés megfogalmazásához a csupa egyes bitből álló ff:ff:ff:ff:ff:ff Ethernet címet használják. Megállapodás szerint az Ethernet alapú hálózatok minden gépe figyeli az ilyen címre küldött csomagokat. Ez azt jelenti, hogy az ARP kérést is látja mindegyikük. Minden egyes gép ellenőrzi, hogy a kérés rá vonatkozik-e. Ha igen, akkor választ küld. Ha nem, akkor egyszerűen nem veszi figyelembe. (Néhány gép az ARP táblázatának frissítésére is használja az ilyen kéréseket, még akkor is, ha az nem rá vonatkozik.) Az üzenetszóró IP csomagokat (pl. 255.255.255.255 vagy 128.6.4.255) is csupa egyes bitből álló Ethernet címre kell küldeni.

10. További információ

Az alábbiakban a főbb protokollokat jellemző dokumentumok felsorolása következik. Mivel többszáz ilyen létezik, ezért csak a legfontosabbnak tűnők szerepelnek a felsorolásban. Az Internet szabványokat RFC-knek hívják, ami a Request For Comments (esetleg Megjegyzést Igénylő Kérés?; erre várom a javaslatokat) kifejezés rövidítése. Ha megszületik egy szabványtervezet, akkor azt először ajánlásként teszik közzé, és kap egy RFC számot. Ha végül az ajánlást elfogadják, akkor Hivatalos Internet Protokoll (Official Internet Protocols) válik belőle, de továbbra is az RFC számmal hivatkoznak rá. A felsorolásba két IEN (Internet Engineering Notes — Internet Műszaki Jegyzet) is bekerült. (Az IEN a hivatalos dokumentumok egy másik osztályozása volt. Ezt ma már nem használják—az összes hivatalos Internet dokumentumot RFC-ként számozzák. A hivatalos írásokra létezik egy levelezési lista is.) Megállapodás szerint minden RFC új számot kap, ha átdolgozzák. Két fontos RFC, az „Internet Számok" (RFC 1166) és a „Hivatalos Internet Protokollok" (RFC 1011) a tartalma miatt nagyon gyakran változik. A legutóbbi verzió száma az rfc-index.txt-ben található meg. A TCP/IP iránt érdeklődőknek javasolt az IP-t leíró RFC 791 tanulmányozása. Az RFC 1812, 1716 és 1009 szintén hasznos lehet. Ezekben az NSFnet által használt átjárók specifikációja, valamint az útvonal-választás szerepel. Mint ilyen, rengeteg, TCP/IP technológiával kapcsolatos részt tartalmaz. Érdemes áttanulmányozni legalább egy alkalmazói protokollt, hogy érezzük a dolog gyakorlati részét is. Erre talán a legjobb a levelezés leírása (RFC 821/822). A TCP (RFC 793) persze alapműnek számít. A specifikáció eléggé összetett, így ennek tanumányozása csak akkor javasolt, ha elég idő és türelem áll rendelkezésünkre a figyelmes olvasáshoz. Szerencsére Jon Postel, a főbb RFC-k szerzője, nagyon jól ír. A TCP RFC-t sokkal könnyebb olvasni, mint ahogy azt a tartalma alapján gondolnánk. Idővel a többi RFC-t is bátran nézegessük.

Következzen tehát a felsorolás:

rfc-index.txt	az összes RFC listája
rfc1122/3	Követelmények az Internet hosztok felé. Több protokoll áttekintése. A protokollok gyengéinek, a gyártók által elfogadott konvencióknak, a gyakorlatban fellépő problémáknak, a problémák megoldásainak a listája. Egy adott protokoll tanulmányozása során ne felejtsük el figyelmesen átnézni, mert a protokollokat leíró rfc-k ezeket az információkat nem tartalmazzák. Ugyanez vonatkozik az rfc1009-re is.
rfc1012	az RFC-k teljesebb listája
rfc1011	Hivatalos Protokollok. Hasznos az átböngészése, hiszen itt olvasható, hogy milyen feladatot látnak el az egyes protokollok. Leírja továbbá, hogy melyik RFC vált szabvánnyá.
rfc1010	Kiosztott Számok. Az Internet-tel dolgozva gyakran lehet erre referenciaként szükség. Olvasni nem olyan izgalmas. A hivatalosan definiált jól-ismert számokat és egyebeket listázza. A legutóbbi változata az rfc1700 Internet Számok nevet viseli.
rfc1009	Követelmények az Internet Átjárók felé. Jól használható bevezetést nyújt az IP útvonal-választáshoz és az átjárókhoz. (Lásd még: rfc1716, rfc1812.)
rfc1001/2	netBIOS: hálózattervezés PC-vel
rfc973	tartományok aktualizálása. Ezen a téren sok új információ jelent meg. Az rfc1034 és rfc1035 újabb verziót jelölnek. Ezek aktualizálása az rfc1101, rfc1876 és az rfc1348, rfc1637, rfc1706.
rfc959	FTP (állományátvitel)
rfc950	alhálózatok
rfc937	POP2: levelek olvasása PC-n
rfc894	IP továbbítása Ethernet-en, lásd az rfc826-t is
rfc882/3	tartományok ('hosztnév <--> IP cím' megfeleltetés, UUCP). Lásd még: rfc973.
rfc854/5	telnet—a távoli bejelentkezés protokollja
rfc826	ARP—Ethernet címek leképezési protokollja (IP címre)
rfc821/2	levelezés—ennek legutóbbi verziója az rfc1495. (Lásd még: rfc987, rfc1148, rfc1327 és rfc1026, rfc1138.)
rfc814	nevek és port-ok—általában az ismertebb port-okról
rfc793	TCP
rfc792	ICMP

rfc791	IP
rfc768	UDP
rip.doc	a legjobban elterjedt útvonal-választási protokoll részletei (--> RFC 1058)
ien-116	régebbi névkiszolgáló (pár rendszer még használja)
ien-48	Catenet modell, a TCP/IP mögötti filozófia általános ismertetése

A következő dokumentumok egy-egy szűkebb területre specializálódtak:

rfc813	TCP ablak, és nyugtázási stratégiák
rfc815	datagramm összerakási technikák
rfc816	hibakizárási és -feloldási módszerek
rfc817	modularitás és hatékonyság az implementációkban
rfc879	a TCP maximális szegmensméret opciója
rfc896	torlódásszabályozás
rfc827,888,904,975,985	EGP (Exterior Gateway Protocol) és azzal kapcsolatos témák
rfc968	A 'Twas the Night Before Start-up című szellemes verset olvashatjuk, melyben a szerző a hálózatok telepítésekor felbukkanó problémákat ecseteli.

A legfontosabb RFC-k három kötetes gyűjteménye a DDN Protocol Handbook (DDN Protokoll Kézikönyv, 1985; ~12 cm vastag), amely a DDN Network Information Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California 94025 (telefon: ++1-800-235-3155) címen rendelhető. Az RFC-k anonim FTP-vel is elérhetők a NIC.DDN.MIL címen. A dokumentumok nevei:

```
RFC:
      /rfc/rfc-index.txt
      /rfc/rfcN.txt, ahol N a kért RFC száma
```

Ajánlott még az InterNIC Directory and Database Services, ds.internic.net kiszolgáló anonim FTP elérése. A keresett RFC dokumentumok az rfc/rfc####.txt vagy rfc/rfc###.ps nevek alatt találhatóak, ahol a #### a kért RFC száma (kezdő nullák nincsenek benne). Ugyanezen kiszolgálótól levélben is kérhető a szolgáltatás. A mailserv@ds.internic.net címre az alábbi üzenetet kell küldeni:

```
document-by-name rfcNNNN
```

Itt az NNNN a kért rfc száma. Amennyiban postscript formátumban kell a szöveg, akkor a

```
document-by-name rfcNNNN.ps
```

üzenetet kell küldeni. Több RFC esetén azokat vesszővel válasszuk el, vagy minden kérést új sorba írjunk. Pl.:

```
document-by-name rfc1791, rfc1792
```

vagy

```
document-by-name rfc1791
document-by-name rfc1792
```

A rip.doc anonim FTP-vel letölthető az src.doc.ic.ac.uk címről /rfc/rip.doc néven. Ajánlatos az ftp://src.doc.ic.ac.uk/rfc/ címen rendesen körülnézni, mert rengeteg, hálózattal kapcsolatos dokumentáció található itt. Vigyázat: a könyvtár listája nagyon hosszú !

Magyarországon az ftp://sunserv.kfki.hu/pub/documents/rfc/ címen érhetők el a különböző rfc dokumentumok.

11. Irodalom

Az RFC-k mellett az alábbi könyvek nagy része ajánlott azoknak, akik (jobban) el szeretnének mélyülni a hálózati protokollokban és a bennük rejlő lehetőségekben.

- Comer, Douglas E.: Internetworking with TCP/IP: Volume I; Principles, protocols, and architecture, 2nd edition, Prentice-Hall International Editions, 1991
- Comer, Douglas E., Stevens, David L.: Internetworking with TCP/IP: Volume II; Design, implementation, and internals, 2nd edition, Prentice-Hall International Editions, 1994
- Comer, Douglas E., Stevens, David L.: Internetworking with TCP/IP: Volume III; Client-server programming and applications, BSD socket version, Prentice-Hall International Editions, 1993

  A fenti háromkötetes mű nagyon értékes információkat tartalmaz a megvalósításokhoz is. Egy sor példaprogramot és kitűzött feladatot ad. Annak ellenére, hogy a TCP/IP 1990 körüli pillanatképét adja, határozottan kézbe kell venni. A szerzők egyébként oktatási segédletként is ajánlják.
- Csórián Sándor: Számítógépes kommunikáció, Cédrus Kiadó, 1993

  Ezt a könyvet elsősorban azoknak érdemes elolvasni, akik a számítógépes kommunikáció terén alapvető ismeretekre szeretnének szert tenni. A szerző az alapokról ír közérthetően.
- Jodál Endre: Adatkommunikáció és számítógép-hálózatok, Cédrus Kiadó, 1993
- Jodál Endre: Informatikai alapszókincs: angol-magyar szótár, Cédrus Kiadó, 1993

  Ezen utóbbi két könyv lexikon szinten használható, főleg angol-magyar számítástechnikai (értelmező) szótárak.
- Quarterman, John S.: The Matrix: computer networks and conferencing systems worldwide, Digital Equipment Corporation, 1990

A Mátrix, azaz az Internet fejlődésének általános leírása. Az egyes országokra, földrészekre lebontva adja meg a Hálózat fejlettségi szintjét, elérhetőségét. Mindezt persze csak 1990-ig bezárólag. Rengeteg referenciát jelöl meg.

- Tanenbaum, Andrew S.: Számítógép-hálózatok, NOVOTRADE Kiadó Kft., 1995

  Az ISO OSI modell hét rétegén keresztül mutatja be a számítógépes hálózatokat. Részletes és átfogó kép. Minden egyes rétegre példákat hoz a megvalósításukkal kapcsolatban. Szintén feladatokat tűz ki minden fejezet végén.