Joe Casad

Tanuljuk meg a TCP/IP használatát 24 óra alatt!

Negyedik kiadás

SAMS KISKAPU

A fordítás a következő angol eredeti alapján készült:

Joe Casade: SAMS Teach Yourself TCP/IP in 24 Hours

Authorized translation from the English language edition, entitled SAMS TEACH YOURSELF TCP/IP IN 24 HOURS, 4th Edition, ISBN 0672329964, by CASAD, JOE, published by Pearson Education, Inc., publishing as Sams Publishing.

Lektorálás: *Dr. Büki András, Rézműves László*
Fordítás: *Dr. Büki András, Rézműves László, Szabó Zoltán*
Műszaki szerkesztő: *Csutak Hoffmann Levente*
Tördelés: *Kis Péter*

Felelős kiadó a Kiskapu Kft. ügyvezető igazgatója
© 2010 Kiskapu Kft.
1134 Budapest, Csángó u. 8.
Fax: (+36-1) 303-1619
http://www.kiskapukiado.hu/
e-mail: kiado@kiskapu.hu

ISBN: 978 963 9637 68 9

Készült a debreceni Kinizsi Nyomdában
Felelős vezető: *Bördős János*

Tartalomjegyzék

A szerzőről

Joe Casad mérnök, író és szerkesztő, aki szerzőként vagy társszerzőként már 12 könyvet írt a számítógépes hálózatok, illetve a rendszerfelügyelet témakörében. Korábban a *C/C++ Users Journal* szerkesztőjeként és a UnixReview.com vezető szerkesztőjeként dolgozott.

Ajánlás

A három tapsoló kéznek
– Joe Casad

Köszönetnyilvánítás

Szeretnénk köszönetet mondani Trina MacDonaldnak, Michael Thurstonnak, Betsy Harrisnek és Ravi Prakash-nak a türelmükért és jó tanácsaikért. Ezen kívül hálával tartozom a következő személyeknek, akik a munkájukkal hozzájárultak a *Tanuljuk meg a TCP/IP használatát 24 óra alatt!* korábbi kiadásainak elkészítéséhez: Bob Willsey, Sudha Putnam, Walter Glenn, Art Hammond, Jane Brownlow, Jeff Koch, Mark Renfrow, Vicki Harding, Mark Cierzniak, Marc Charney és Jenny Watson.

Mondja el a véleményét!

Az *olvasó* a legfontosabb kritikus, akinek a véleménye nekem és a kiadónak is roppant értékes. Szeretnénk tudni, mit csinálunk jól, mi az, amin javíthatnánk, milyen könyveket kellene megjelentetnünk, és így tovább. *Felhívjuk a figyelmet, hogy a könyv témájával kapcsolatos szakmai kérdésekben nem tudunk segíteni, és a nagy számban beérkező levelek miatt nem biztos, hogy minden üzenetre válaszolunk.*

Kérjük, ha ír, tüntesse fel a könyv szerzőjét és címét, valamint a saját nevét, telefonszámát vagy e-mail címét. Megjegyzéseit alaposan áttanulmányozzuk, és továbbítjuk a könyv szerzőjének és szerkesztőinek.

E-mail: networking@samspublishing.com
Levél: Mark Taub
Editor-In-Chief
Sams Publishing
1330 Avenue of the Americas
New York, NY 10019 USA

Bevezetés

Üdvözli a *Tanuljuk meg a TCP/IP használatát 24 óra alatt!* (*Sams Teach Yourself TCP/IP in 24 Hours, negyedik kiadás*). Ez a könyv világos és tömör bevezetést nyújt azoknak, akik most ismerkednek a TCP/IP-vel, de azok is haszonnal forgathatják, akik már dolgoztak a TCP/IP-vel, de egy kicsit többet szeretnének tudni róla. Ez a kiadás a TCP/IP újabb fejlesztéseihez igazodva új anyagokkal gyarapodott, és az alábbi témákkal is közelebbről foglalkozik:

- Tűzfalak
- Áramló tartalmak
- Webszolgáltatások

Az áramló tartalmakkal, a webszolgáltatásokkal és az új Webbel új fejezetek foglalkoznak, míg a TCP/IP egyéb újdonságairól a könyv új szakaszai ejtenek szót.

Minden fejezet feldolgozása egy óráig tart?

A fejezeteket úgy szerkesztettük meg, hogy a fogalmak egy óra alatt elsajátíthatók legyenek, és minden fejezet elég rövid ahhoz, hogy egyhuzamban végig lehessen olvasni. Egy-egy fejezettel valójában kevesebb mint egy óra alatt végezni lehet, így az egy órába az is belefér, hogy jegyzeteljünk, és újraolvassuk a bonyolultabb részeket.

Hogyan használjuk a könyvet?

A *Sams Teach Yourself* (*Tanuljuk meg...*) sorozatot arra tervezték, hogy az Olvasó néhány könnyű és olvasmányos óra alatt képes legyen megtanulni egy-egy témakört. A *Tanuljuk meg a TCP/IP használatát 24 óra alatt!* hat részre oszlik, amelyek mindegyike egy lépéssel közelebb visz minket ahhoz, hogy mesteri szinten elsajátítsuk a TCP/IP használatát.

- Az I. rész (A TCP/IP alapjai) bevezetést nyújt a TCP/IP és a TCP/IP protokollverem világába.
- A II. rész (A TCP/IP protokollrendszere) a TCP/IP egyes protokollrétegeit veszi górcső alá: a hálózati hozzáférés, az Internet, az átvitel (szállítás) és az alkalmazások rétegét. Tanulunk az IP címzési rendszeréről és az alhálózatokról, valamint a fizikai hálózatokról és az alkalmazásszolgáltatásokról. Ezen kívül megismerjük azokat a protokollokat is, amelyek a TCP/IP egyes rétegeiben működnek.

- A III. rész (Hálózatkezelés a TCP/IP segítségével) a TCP/IP-hálózatok támogatásához szükséges eszközök, szolgáltatások és segédprogramok közül mutat be néhányat, valamint szót ejt az útválasztó és hálózati hardvereszközökről, a DHCP-ről, a DNS-ről és az IPv6-ról.
- A IV. rész (TCP/IP-eszközök) azokat a leggyakrabban használt eszközöket mutatja be, amelyeket a TCP/IP-hálózatok beállítására, kezelésére és hibaelhárítására használnak. Hallunk majd a Pingről, a Netstatról, az FTP-ről, a Telnetről és más hálózati segédprogramokról.
- Az V. rész (A TCP/IP és az Internet) a világ legnagyobb TCP/IP-hálózatát, az Internetet mutatja be. Megismerjük az Internet felépítését, valamint a HTTP-t, a HTML-t, az XML-t, az elektronikus levelezést és az áramló internetes tartalmakat.
- A VI. rész (Haladó témák) olyan témakörökkel foglalkozik, mint a webszolgáltatások, az üzenetküldés, a jelentésközpontú Web és a TCP/IP biztonsága. A VI. rész egy esettanulmánnyal zárul, amelyben azt mutatjuk be, hogy a TCP/IP összetevői hogyan működnek együtt egy valós munkakörnyezetben.

A könyvben tárgyalt fogalmak és eljárások magához a TCP/IP-hez hasonlóan rendszerfüggetlenek, és az RFC-dokumentumokban (Internet Requests for Comment) meghatározott szabványokból erednek.

A könyv fejezeteinek szerkezete

A *Tanuljuk meg a TCP/IP használatát 24 óra alatt!* minden órája egy rövid bevezetéssel, valamint az óra céljainak felsorolásával kezdődik. Emellett minden fejezetben megtaláljuk az alábbi részeket:

Törzsszöveg

Minden órának van egy törzsszövege, amely világos, közérthető formában tárgyalja a fejezet témáját. A szövegben leírt fogalmak magyarázatát ábrák és táblázatok segítik, valamint a szövegben elszórva külön jelzéssel ellátott megjegyzéseket is találunk, amelyek meghatározásokat, leírásokat vagy figyelmeztetéseket tartalmaznak, amelyek az anyag jobb megértését szolgálják.

Ezek a keretes megjegyzések a törzsszövegben tárgyalt fogalmakat igyekeznek világosabbá tenni. A megjegyzésekben kiegészítő információkat vagy példákat is találhatunk, de az elolvasásuk jellemzően nem létfontosságú az adott téma megértéséhez. Ha sietünk, vagy csak az alapszintű tudnivalókra vagyunk kíváncsiak, az így jelölt megjegyzéseket átugorhatjuk.

Kérdezz–felelek

Minden óra kérdésekkel zárul, amelyek a fejezetben tanultak mélyebb megértését és ellenőrzését szolgálják. A kérdésekhez a válaszokat is mellékeltük.

Bizonyos fejezetekben gyakorlatokat is találunk, amelyek a részletek elsajátítását vagy egy adott feladat elvégzésének gyakorlását segítik. Csak azokhoz az órákhoz mellékeltük őket, ahol a gyakorlati feladatok segíthetnek jobban megérteni az anyagot. Még ha nem is rendelkezünk a gyakorlatok némelyikének elvégzéséhez szükséges szoftverrel vagy hardverrel, akkor is érdemes elolvasni őket, hogy lássuk, hogyan működnek az eszközök egy valódi hálózatban.

Kulcsfogalmak

Minden fejezetben összefoglaljuk az órán megismert legfontosabb fogalmakat. Ezeket a kulcsfogalmakat mindig az óra végén találjuk, ábécérendbe szedve.

I. RÉSZ

A TCP/IP alapjai

1. ÓRA

A TCP/IP áttekintése

Ebben az órában a következőkről lesz szó:

- Hálózatok és hálózati protokollok
- A TCP/IP protokoll története
- A TCP/IP fontosabb szolgáltatásai

A TCP/IP egy protokollrendszer, vagyis hálózati kommunikációra szolgáló protokollok gyűjteménye. Ha válaszolni akarunk arra a kérdésre, hogy tulajdonképpen mi is az a „protokoll", akkor először egy másik kérdést kell föltennünk: mi az, hogy „hálózat"?

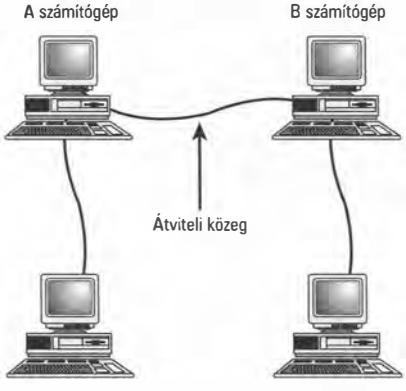
Ebben az órában először is megtudjuk, mi a válasz ez utóbbi kérdésre, majd az is kiderül, hogy egy hálózatnak miért van szüksége protokollokra. Azt is megtudjuk, hogy mi is az a TCP/IP, mire jó, és hol kezdődött a története.

Aki elsajátítja ennek az órának az anyagát, az képes lesz válaszolni a következő kérdésekre:

- Mi a hálózat definíciója?
- Mire való egy hálózati protokollgyűjtemény?
- Mi az a TCP/IP?
- Milyen fejlődési lépéseken ment keresztül a TCP/IP protokoll?
- Milyen fontosabb szolgáltatásai vannak a TCP/IP protokollnak?
- Milyen szervezetek felügyelik a TCP/IP fejlesztését illetve az internet működését?
- Mik azok az RFC-k és hol találjuk meg őket?

Hálózatok és protokollok

A hálózat nem más, mint számítógépek vagy hozzájuk hasonló eszközök olyan csoportja, amelyek egymással egy közös átviteli közegen keresztül kommunikálnak. Ezt szemlélteti sematikusan az 1.1. ábra.

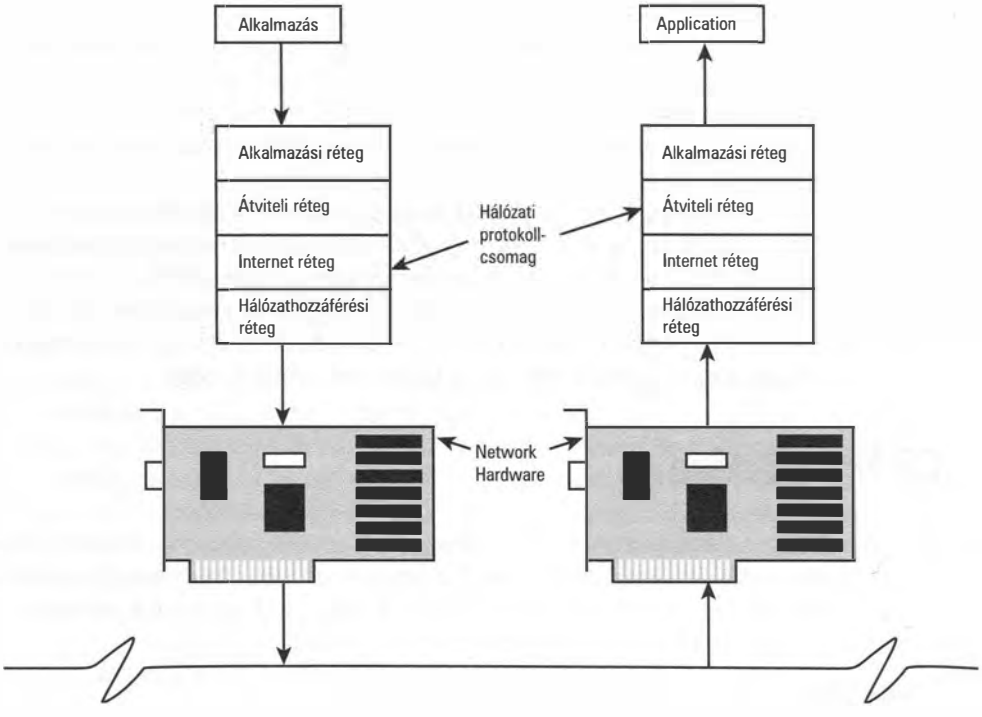


1.1. ábra
Egy tipikus helyi hálózat

Egy hálózatban a kiszolgálásra irányuló kérések, illetve a válaszul küldött adatok egy átviteli közegen keresztül jutnak el az egyik számítógéptől a másikig. (Ami az átviteli közeget illeti, az lehet egy hálózati kábel, egy telefonvonal vagy a mai elterjedt drótnélküli kapcsolatok esetében rádióhullámok.) Az 1.1. ábrán bemutatott hálózatban az A jelű számítógépnek képesnek kell lennie arra, hogy kérést vagy üzenetet küldjön a B gépnek, B-nek pedig képesnek kell lennie arra, hogy megértse A üzenetét és hogy megfelelő módon válaszolni tudjon A kérésére.

Egy számítógép egy vagy több alkalmazáson keresztül tartja a kapcsolatot a külvilággal. Ezek a kommunikációs alkalmazások azok, amelyek ellátják a különféle feladatokat, és kezelik a be- és kimeneti műveleteket. Ha az illető számítógép egy hálózat része, akkor egy vagy több olyan alkalmazással is rendelkeznie kell, amelyek képesek más számítógépekkel kommunikációt folytatni a hálózaton keresztül. A **hálózati protokoll** olyan egyezményes szabályok gyűjteménye, amelyek meghatározzák, miként kell megvalósítani a gépek közt az adatátvitel bonyolult műveletét. Adatátvitelnél az egyik gépen futó alkalmazás által elküldött adatok először a gép hálózati hardverén keresztül a kommu-

nikációs csatornába jutnak, megérkeznek a megfelelő címzetthez, majd annak a gépnek a hálózati hardverén keresztül végül eljutnak a másik gépen futó fogadó alkalmazáshoz (lásd az 1.2. ábrát).



1.2. ábra

Egy hálózati protokollcsomag funkciói

A TCP/IP szabvány célja az, hogy biztosítsa a kompatibilitást az egyes TCP/IP megvalósítások között függetlenül azok készítőjétől vagy verziószámától.

A TCP/IP rendszert alkotó protokollok pontosan leírják, miként kell megvalósítani a hálózati kommunikációt két gép között, és ami még ennél is fontosabb meghatározzák, hogyan kell kinéznie egy adategységnek, és milyen kiegészítő információkat kell tartalmaznia ahhoz, hogy a fogadó gép megfelelően tudja dekódolni a neki küldött üzenetet. A TCP/IP és a vele kapcsolatos egyéb protokollok olyan teljes rendszert alkotnak, amely részletesen meghatározza, miként kell az adatokat elküldeni, feldolgozni, továbbítani és fogadni egy hálózatban. A protokollok egy ilyen teljes rendszerét **protokollcsomagnak** (*protocol suite*), vagy protokollveremnek nevezzük.

Az a szoftvercsomag, amely egy konkrét rendszeren a TCP/IP-n alapuló kommunikációt bonyolítja, vagyis megformázza és feldolgozza az átvitt adatokat a TCP/IP egy adott **megvalósítása** (*implementation*), vagy implementációja. Példának okáért egy Microsoft

Windows operációs rendszert futtató számítógép képes részt venni egy TCP/IP hálózatban, ami annak köszönhető, hogy a Microsoft elkészítette e protokollcsomag egy megvalósítását. A könyv olvasása során fontos különbséget tenni a következő két fogalom között:

- A **TCP/IP egy szabvány**, vagyis szabályok olyan gyűjteménye, amelyek alapján egy TCP/IP hálózat működik.
- A **TCP/IP egy megvalósítása** ezzel szemben olyan szoftverkomponens, amely lehetővé teszi, hogy egy számítógép részt vegyen egy TCP/IP hálózat kommunikációjában.

A fent vázolt, és amúgy igen lényeges különbségtétel a TCP/IP szabvány és annak megvalósításai között a közbeszédben gyakran elmosódik, és ez sokszor összezavarja a gyanútlan olvasókat. Egyes szerzők például előszeretettel beszélnek a TCP/IP modell rétegeiről, valamint azok szolgáltatásairól, amelyeket más rétegek számára nyújtanak. A helyzet az, hogy a TCP/IP modell csak definiálja de nem nyújtja ezeket a szolgáltatásokat. A szolgáltatás már a konkrét megvalósítás dolga.

A TCP/IP fejlődése

A TCP/IP jelenlegi fejlettségi szintjén tulajdonképpen két fejlesztési projekt találkozásának az eredménye. Mindkettő az 1970-es években kezdődött, és elmondható róluk, hogy egyesült erővel néhány évtized alatt teljesen átalakították a számítástechnika világát. A két nagyszabású projekt a következő volt:

- Az internet
- A helyi hálózat

Az internet

A TCP/IP felépítése ma is alapvetően azt a történelmi szerepet tükrözi, amelyet eredetileg szántak neki, vagyis hogy ez legyen az internet kommunikáció protokollja. Az internet, mint megannyi más fejlesztési projekt az Egyesült Államok Védelmi Minisztériumában kezdődött még valamikor az 1960-as évek második felében. A védelmi szakembereknek ebben az időben tűnt fel, hogy a hadsereg számos különböző telephelyen meglehetősen sok és sokféle számítógépet halmozott fel, amelyek azonban vagy egyáltalán nem voltak hálózatba kapcsolva, vagy olyan egyedi (*proprietary*) módszerekkel voltak összekötve, amelyek kizárták a különböző rendszerek közti együttműködést.

Az **egyediség** ebben az összefüggésben azt jelenti, hogy a kérdéses kommunikációs technológiát egy cég fejlesztette és felügyelte, amely cégnek számos esetben nem állt érdekében, hogy kellő mennyiségű információt szolgáltasson a protokollokról ahhoz, hogy a felhasználók össze tudják kapcsolni a különböző rendszereket.

A védelmi szakemberek ezt látva természetesen elkezdtek azon töprengeni, miként lehetne mégis megvalósítani az összeköttetést a különböző rendszerek között. Mivel pedig az alapvető foglalkozásuk a védelem volt, nem meglepő, hogy rögtön rájöttek: ha meg is épülne egy ilyen rendszer, az garantáltan célpontot jelentene az ellenség számára. Ha tehát ragaszkodnak az elképzeléshez, akkor a megépülő hálózat egyes számú tulajdonsága a decentralizáltság kell legyen. Ez azt jelenti, hogy az alapvető hálózati szolgáltatások nem futhatna egy vagy néhány sebezhető számítógépen. Mivel azonban a rakéták korában minden rendszer sebezhető, úgy döntöttek, olyan hálózatot fognak építeni, amelyben egyáltalán nincsenek „hibapontok". Egy ilyen rendszernek bármely részére dobnak is bombát, a maradék működőképes marad. Így született meg az ARPAnet, amelynek neve a minisztérium kutatási részlegének rövidítése (Defense Department Advanced Research Projects Agency).

Amint ez a hálózat kezdett alakot ölteni, tudósok egy csoportja Robert E. Kahn és Vonton Cerf vezetésével elkezdett egy olyan protokollcsomagon kifejlesztésén dolgozni, amely rugalmasan, redundáns és decentralizált módon képes eljuttatni akár nagy adatmennyiségeket is bárhonnan bárhova. Ennek a munkának az eredménye lett aztán a TCP/IP alapja. Amikor később az NSF (National Science Foundation) olyan hálózatot akart építeni, amelyek a kutatóhelyeket köti össze, adoptálták az ARPAnet protokollrendszerét és elkezdték megépíteni azt a dolgot, amit ma úgy hívunk hogy internet. A TCP/IP kezdeti fejlesztésében a University College of London és más európai intézmények is részt vettek, így 1975 körül megkezdődtek az első kontinensközi kommunikációs tesztek is.

Amint az ebből a könyvből is ki fog derülni, az ARPAnet decentralizáltságra való törekvése ma is világosan megfigyelhető a TCP/IP protokollok tervezésén és működésén. Amúgy az internet óriási sikerének egyik záloga éppen ez a decentralizált viselkedés volt. A TCP/IP decentralizáltságát alapvetően két tulajdonsága biztosítja:

- **Ellenőrzés a végpontokon** – Végpontoknak nevezzük azt a két számítógépet, amelyek kommunikálnak egymással. Az elnevezés onnan ered, hogy magában az adatátvitelben általában más számítógépek egész sora vesz részt, e kettő ennek a láncolatnak a két végpontja. A TCP/IP esetében az átvitt adatok ellenőrzéséért és visszaigazolásáért a végpontok a felelősek. E tekintetben a hálózat valamennyi gépe egyenrangúnak számít, és nincs semmiféle központilag előírt séma, ami alapján központilag kellene felügyelni a kommunikációt.
- **Dinamikus útválasztás** – A hálózat bármely két végpontját általában több lehetséges továbbítási útvonal köti össze, a ténylegesen használandót pedig ezek közül az útválasztók jelölik ki a hálózat pillanatnyi állapotának megfelelően. Az útválasztásról és a lehetséges útvonalak kiválasztásáról a későbbiekben még részletesen esik szó.

A helyi hálózat (LAN)

Amint az internet elkezdett kiépülni az egyetemek és kutatóintézetek körül, megjelent egy új hálózati fogalom is, mégpedig a helyi hálózat (Local Area Network; LAN) fogalma. A helyi hálózatok magával a számítástechnikával párhuzamosan folyamatosan fejlődtek amit alapvetően az az igény vezérelt, hogy a különböző irodák kezdettől fogva szerették volna egymással megosztani számítástechnikai erőforrásaikat.

A helyi hálózatok korai formái általában nem biztosítottak hozzáférést az internethez és egyedi protokollokat használtak. A legtöbb ilyen hálózatnak amúgy eleve nem is létezett olyan szolgáltatása, amivel útválasztást lehetett volna megvalósítani. Végül aztán megjelentek az olyan vállalatok, amelyek mindenképpen össze szerették volna kapcsolni az egymással inkompatibilis helyi hálózataikat, és erre a TCP/IP protokollt találták a legalkalmasabbnak. Ahogy terjedt az internet, egyre növekedett azoknak a felhasználóknak a száma is, akik munkájuk során használni szerették volna ezt a kommunikációs csatornát. Először természetesen itt is mindenféle egyedi megoldások jelentek meg arra, miként lehet a LAN-okat az internet részévé tenni. Legtöbbször speciális átjárókkal gondoskodtak az eléréshez szükséges protokollfordításról, később aztán a hálózati szoftverek fejlesztői elkezdtek többé-kevésbé teljes megoldásokat kínálni az internetkapcsolat kiépítésére. A Mac OS és a Windows legújabb változatai már olyannyira támogatják a TCP/IP-t, hogy maguknak a helyi hálózatoknak is ez lett az alapértelmezett kommunikációs protokollja. A TCP/IP ezzel együtt alapvetően a Unix rendszerek körül kezdett el fejlődni, így természetes, hogy valamennyi Unix változat „folyékonyan beszéli" a TCP/IP-t. Mivel pedig az utóbbi időkben megugrott az olyan Unix alapú rendszerek népszerűsége, mint a Linux, a BSD, a Solaris vagy az Apple OS X, a TCP/IP dominanciája még tovább növekedett a hálózatok világában.

Az „átjáró" (*gateway*) kifejezést a TCP/IP-vel kapcsolatban meglehetősen inkonzisztens módon használják. Az átjáró sok esetben nem egyéb, mint egy olyan közönséges útválasztó, amely kapcsolatot létesít egy helyi hálózat és egy nagyobb hálózat között (bővebben lásd az átválasztásról szóló részt ennek az órának az anyagában). Ugyanakkor maga a kifejezés takarhat olyan speciális képességekkel rendelkező útválasztó eszközt is, amely protokollok közti tolmácsolást végez, nem csak egyszerű forgalomirányítást.

Amint arról a harmadik órában bővebben is lesz majd szó, a helyi hálózatok fejlődése számos olyan, a hálózati hardverekkel kapcsolatos protokoll kifejlesztését indította el, amelyek ma szilárd alapot jelentek a TCP/IP számára.

A TCP/IP szolgáltatásai

A TCP/IP protokollcsomagnak számos olyan fontos szolgáltatása van, amelyekről szó lesz ebben a könyvben. A legfontosabbak ezek közül talán a következők:

- Logikai címzés
- Útválasztás
- Névfeloldás
- Hibakezelés és folyamatszabályozás
- Alkalmazások támogatása

Ezek a szolgáltatástípusok alkotják a TCP/IP lényegét. A következő szakaszokban egyenként is megvizsgáljuk valamennyit, a könyv későbbi fejezeteiben pedig részletesen is szó esik majd róluk.

Logikai címzés

Valamennyi hálózati csatolónak van egy egyedi és megváltoztathatatlan fizikai címe. A fizikai cím (amit egyes esetekben MAC címként is említenek) egy szám, amit a gyárban rendelnek hozzá az adott hálózati kártyához. Egy helyi hálózatban alacsony szintű, a hardverhez viszonylag közel működő protokollok gondoskodnak az adatok fizikai közegen való átviteléről, melynek során a hardvercímet használják a gépek azonosítására. Számos különböző hálózattípus létezik, és mindegyik más módszert használ az adatok továbbítására. Egy közönséges Ethernet hálózatban például a számítógép az adatokat közvetlenül az átviteli közegbe juttatja. A hálózati csatolók valamennyi ilyen adást figyelik, és kiválasztják közülük azt, amelyik az ő saját fizikai címüket tartalmazza, vagyis nekik szól.

Amint arról a 9. órában részletesebben is lesz majd szó, a mai Ethernet hálózatok a fent leírt idealizált megvalósításnál, melyben a számítógép közvetlenül írja az adatokat a kommunikációs csatornába egy kissé bonyolultabbak is lehetnek. Számos hálózat tartalmaz például olyan hardvereszközöket, például kapcsolókat, amelyek kezelik az átvinni kívánt jeleket.

A nagy hálózatokban természetesen nincs mód arra, hogy valamennyi számítógép valamennyi átvitelre figyeljen, és kiválassza a neki szóló adásokat (gondoljunk csak bele mi történne, ha a saját asztali gépünk a világ összes adatátvitelét hallgatná, miközben az interneten böngészünk). Minél több számítógép csatlakozik egy fizikai átviteli közegre, annál nagyobb problémába ütközik és annál kevésbé hatékony a fizikai címzésen alapuló kommunikáció. A hálózati adminisztrátorok gyakran különféle hálózati eszközök, például útválasztók segítségével szegmentálják a nagyobb hálózatokat, hogy

csökkentsék azok forgalmát. Az útválasztót is tartalmazó nagyobb hálózatokban az adminisztrátorok gyakorlatilag mindig kisebb **alhálózatokat** (*subnet*) jelölnek ki. Ezek egy logikai hierarchiát alkotnak, így gondoskodnak arról, hogy az üzenetek hatékony módon juthassanak el a címzetthez. A TCP/IP protokoll az alhálózatok kialakítását az úgynevezett **logikai címzésen** (*logical addressin*) keresztül támogatja. A logikai cím olyan cím, amit a hálózati szoftver segítségével rendelünk hozzá a hálózat egy elemé-hez. A TCP/IP esetében a logikai címeket **IP címeknek** (*IP address*) nevezzük. Amint arról a 4. és 5. órában még részletesen szó lesz, egy IP cím a következőket tartalmazhatja:

- Egy hálózati azonosítószámot (*network ID*)
- Egy alhálózati azonosítószámot (*subnet ID*), ami az adott hálózat egy alhálózatát jelöli ki
- Egy számítógép azonosítót (*host ID*)

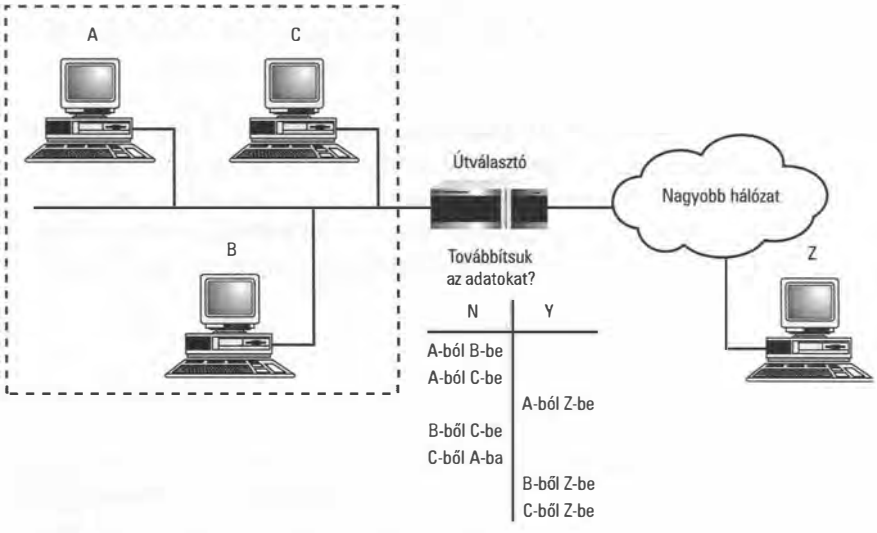
Az IP címek rendszere lehetővé teszi, hogy a hálózati adminisztrátorok olyan logikus címzési rendszert alakítsanak ki egy hálózaton belül, amely a címek egymásutániságával világosan tükrözi a szervezeti felépítést, vagy magának a hálózatnak a logikai struktúráját.

Ha hálózatunk nem kapcsolódik az internethez, akkor gyakorlatilag olyan IP címeket használunk, amilyeneket csak akarunk. Kizárólag az IP címzéssel kapcsolatos leg-alapvetőbb szabályokat kell szem előtt tartanunk. Ha azonban hálózatunk az internet része, akkor az 1988-ban alapított ICANN (Internet Corporation of Assigned Names and Numbers) nevű szervezettől igényelnünk kell egy hálózati azonosítót, amely valamennyi hálózati címünk előtagja lesz (erről a 4. és 5. órában lesz részletesebben szó). Létezik ugyanakkor egy érdekes fejlesztés ezzel kapcsolatban, az úgynevezett címfordítás (Network Address Translation; NAT), amely azt teszi lehetővé, hogy útválasztók által nem kezelt, úgynevezett privát címeket használjunk saját hálózatun-kon belül, majd ezeket valódi internet címekké fordítsuk le, ha egy gép az interneten át szeretne kommunikálni. Erről a speciális szolgáltatásról a 12. órában lesz bővebben szó.

A TCP/IP rendszerében a logikai címeket az ARP és a RARP protokollok alakítják fizikai címekké és vissza. Ezekről a 4. órában esik majd szó.

Útválasztás

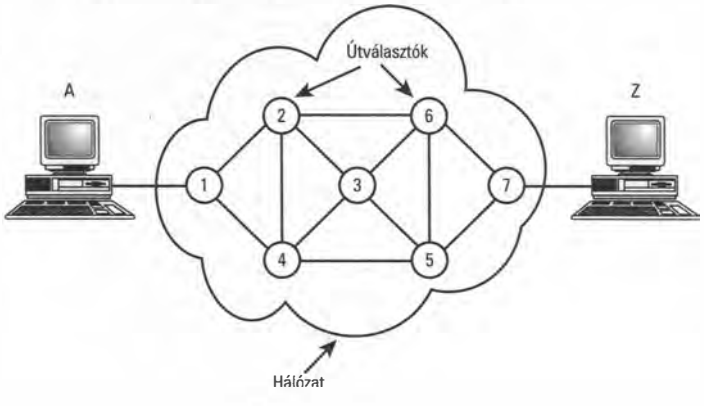
Az **útválasztó** (*router*) olyan speciális eszköz, amely képek kiolvasni a hálózati forgalomban továbbított adatokból a logikai címeket, és ez alapján a megfelelő helyre továbbítani a forgalmat. A legegyszerűbb esetben egy útválasztó egy kisebb alhálózatot köt össze egy nagyobb hálózattal (lásd az 1.3. ábrát).



1.3. ábra

Egy alhálózat és egy nagyobb hálózat összekapcsolása útválasztó segítségével

Azok az adatok, amelyeket a helyi alhálózat egyik gépe küld egy ugyanabban az alhálózatban található másik gépnek, nem jutnak át az útválasztón, így nem zavarják fölöslegesen a nagyobb hálózat kommunikációját. Ha ellenben az üzenet egy az alhálózaton kívül eső gépnek szól, az útválasztó megfelelően továbbítja azt. Amint azt már korábban is említettük, a nagy hálózatok (mint amilyen maga az internet is) számos útválasztót tartalmaznak, és több különböző átviteli utat biztosítanak két pont között (lásd az 1.4. ábrát).



1.4. ábra

Egy útválasztókkal összekapcsolt hálózat

A TCP/IP természetesen tartalmaz olyan protokollokat is, amelyek alapján az útválasztók eldönthetik, milyen úton kell továbbítani egy adott adatfolyamot a hálózaton keresztül. Az útválasztási protokollokról részletesen a 8. órában lesz szó.

Amint arról a 9. órában részletesebben is esik majd szó, az olyan hálózati eszközök, mint a kapcsolók (switch), hidak (bridge) és okos hub-ok (smart hub) szintén képesek szűrni a hálózati forgalmat és ezzel csökkenteni a vonalak terheltségét. Ugyanakkor mivel ezek az eszközök a fizikai és nem a logikai címekkel dolgoznak, nem képesek ellátni azokat az összetett forgalomirányítási funkciókat, amelyeket az 1.4. ábrán szemléltettünk.

Névfeloldás

Bár a numerikus IP-címek sokkal inkább nevezhetők felhasználóbarátnak mint a hálózati csatolók fizikai címei, azért a helyzet ezekkel is az, hogy alapvetően számítógépeknek szánták őket és nem embereknek. Míg egy számítógép emlékezőtehetségével általában semmi gond nincsen, egy embernek nyilván okozhat gondot arra visszaemlékezni, hogy egy adott számítógép címe 111.121.131.146, vagy 111.121.131.156. A TCP/IP éppen ezért egy az IP címek rendszerével párhuzamosan használható, de betűkből álló címzési rendszer használatát is lehetővé teszi. Ezeket az alfanumerikus címeket nevezzük tartományneveknek (domain name), vagy röviden DNS neveknek. Azt a folyamatot, amellyel a tartományneveket IP címekké képezzük le **névfeloldásnak** (*name resolution*) nevezzük. A fordítás alapjául szolgáló táblázatokat speciális számítógépeken, az úgynevezett **névkiszolgálókon** (*name server*) tárolják.

Az olyan általánosan használt címeket, mint amelyeket e-mail írása, vagy webböngészés során használunk, gyakorlatilag mindig DNS nevek formájában adjuk meg (például *www.microsoft.com*, falcon.ukans.edu, vagy idir.net). A TCP/IP névszolgáltatási rendszere egy logikai hierarchiába rendezi a különböző szintű regisztrált névkiszolgálókat. Ez a többlépcsős rendszer gondoskodik arról, hogy az egyszerű felhasználónak gyakorlatilag soha ne kelljen kézzel lefordítania egy DNS nevet IP címmé.

A DNS az egész internet névfeloldási rendszere és ezzel gyakorlatilag a legelterjedtebb névfeloldási rendszer is egyben. Ugyanakkor léteznek más módszerek is az alfanumerikus nevek IP címmé alakítására. Ezek jelentősége az utóbbi években jelentősen csökkent, de sok helyen a mai napig használják például a WINS (Windows Internet Name Resolution) rendszert, amely NetBIOS neveket képes IP címekké feloldani.

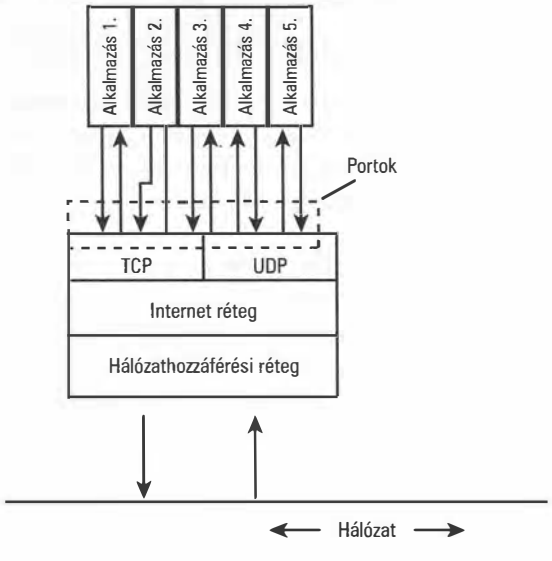
A TCP/IP névfeloldási rendszeréről bővebben a 11. órában lesz majd szó.

Hibakezelés és forgalomszabályozás

A TCP/IP protokollcsomag tartalmaz olyan szolgáltatásokat, amelyek lehetővé teszik a megbízható adatátvitelt a hálózat két pontja között. Ezek gyakorlatilag kétféle műveletet végeznek. Egyrészt lehetővé teszik az átvitt adatok ellenőrzését, mely révén a fogadó meggyőződhet róla, hogy a megérkezett adatok megegyeznek az elküldött információval, másrészt lehetővé teszik, hogy a fogadó visszaigazolja az adatok megérkezését a küldőnek. A TCP/IP adatátviteli rétege (*transport layer*, lásd a 6. óra anyagát) számos hibaellenőrzési, forgalomszabályozó és visszaigazoló műveletet tesz lehetővé a TCP protokoll segítségével. A TCP/IP hálózathozzáférési rétegében (*network access layer*) található alacsonyabb szintű protokollok szintén részt vesznek a hibák kezelésében és javításában.

Alkalmazások támogatása

Egy számítógépen egyszerre több hálózati alkalmazás is futhat. Ennek megfelelően egy protokollcsomagban lennie kell olyan mechanizmusnak, amely alapján eldönthető, hogy egy beérkezett csomagnak pontosan melyik alkalmazás a címzettje. A TCP/IP esetében a hálózat és az alkalmazások ilyen összekapcsolását logikai csatornák, az úgynevezett **portok** (kapuk) segítségével oldják meg. Minden portnak van egy azonosítószáma, amely alapján megcímezhető. A portokat leginkább afféle logikailag definiált csöveknek tekinthetjük, amelyek a számítógép belsejében futva összeköttetést teremtenek a protokollcsomag és a hálózati alkalmazások között, és rajtuk az adatok oda-vissza áramolhatnak (lásd az 1.5. ábrát).



1.5. ábra
Az alkalmazások a hálózathoz logikai csatornákon, az úgynevezett portokon keresztül férhetnek hozzá.

A TCP/IP adatátviteli rétegéhez tartozó TCP és UDP portok részletes tárgyalását a 6. óra anyaga tartalmazza, a hálózati alkalmazások támogatásáról pedig részletesebben a 7. órában esik majd szó, az alkalmazási réteg kapcsán.

A TCP/IP csomag igazából maga is tartalmaz számos olyan használatra kész hálózati alkalmazást, amelyekkel a legalapvetőbb hálózati műveletek elvégezhetők. Ezek közül a legfontosabbak felsorolását az 1.1. Táblázat tartalmazza. A TCP/IP csomagban található segédprogramokról részletesen a 14. órában lesz szó.

1.1. Táblázat *A legfontosabb TCP/IP segédprogramok*

Segédprogram	Felhasználási cél
ftp	Fájlok átvitele
lpr	Nyomtatás
ping	Beállítás/Hibakeresés
route	Beállítás/Hibakeresés
telnet	Távoli terminál
traceroute	Beállítás/Hibakeresés

E könyv írásakor a TCP/IP fejlődése éppen egy új korszakába lép. Az olyan új technológiák, mint a drótnélküli hálózatok, a címfordítás (NAT) vagy a virtuális magánhálózatok növelik a rendszerek összetettségét és olyan új problémákat vetnek fel, amelyekre a TCP/IP tervezői eredetileg nem is számítottak. Az új technológiákról a következő fejezetekben még részletesen esik majd szó.

A szabványosítással foglalkozó szervezetek és az RFC-k

Az internet és a TCP/IP fejlesztésében kezdettől fogva több szervezet vett részt. Ezek közül az egyik – mint arról már szó volt – az Egyesült Államok hadserege volt, ami egyben kiváló magyarázatot szolgáltat arra is, miért burjánzottak el annyira a hálózatokkal kapcsolatban a betűszavak. Akár a TCP/IP múltját, akár a jelenét tekintjük, a következő szervezetek azok, amelyek föltétlen említést érdemelnek vele kapcsolatban:

- IAB (Internet Architecture Board) – Ez egy olyan bizottság, amely az internet és a TCP/IP fejlesztésével kapcsolatos szabályokat alkotja.
- IETF (Internet Engineering Task Force) – Az IAB egyik szárnya, amely tervezéssel kapcsolatos kérdéseket tanulmányoz illetve maga is szabályokat alkot. Az IETF munkacsoportokr aoszlik, amelyek az internet és a TCP/IP fejlesztésének egyes részterületeivel foglalkoznak. Ilyen részterület például az alkalmazások fejlesztése, az útválasztás, vagy a hálózatkezelés.
- IRTF (Internet research Task Force) – Az IAB azon alszervezete, amely a hosszú távú kutatásokat támogatja.

- ICANN (Internet Corporation for Assigned Names and Numbers) – Ez egy 1998-ban alapított szervezet, amely a tárománynevek (domain names), IP címek, valamint a globálisan egyedi protokollparaméterek (ilyenek például a portszámok) kiosztását szabályozza (*www.icann.com*).

A TCP/IP-vel kapcsolatos dokumentáció legnagyobb része bárki számára hozzáférhető az úgynevezett RFC-k (Request For Comment) formájában. Az RFC-k mára egy teljes könyvtárat alkotnak, amelyben megtaláljuk az internettel kapcsolatos szabványokat, illetve a munkacsoportok jelentéseit egyaránt. Magukat az IETF által kibocsátott hivatalos specifikációkat is RFC-k formájában teszik közzé. Sok RFC ugyanakkor nem föltétlen szabványokat tartalmaz, csupán arra hivatott, hogy rámutasson az internet vagy a TCP/IP működésével kapcsolatos egyes problémákra. RFC-t bárki benyújthat. Ennek alapvetően két módja létezik. Küldhetünk ajánlást magának az IETF-nek (proposed RFC), vagy elküldhetjük az anyagot közvetlenül az RFC-k szerkesztőjének (*rfc-editor@rfc-editor.org*).

Az RFC-k az alkotók szándékának megfelelően alapvető információt nyújtanak bárkinek, aki mélyebb ismereteket kíván szerezni a TCP/IP működéséről. A lista meglehetősen tág. Vannak kifejezetten műszaki hangvételű cikkek protokollokról, segédprogramokról és szolgáltatásokról, de találunk itt néhány a TCP/IP-vel kapcsolatos költeményt, illetve pár Shakespeare stílusában megírt művet, amelyek azonban sajnálatos módon sem a világos fogalmazás, sem a gazdaságosság tekintetében nem veszik fel a versenyt a TCP/IP szellemével.

Az RFC-k az interneten több helyen is megtalálhatók. Az első hely természetesen a *www.rfc-editor.org*. A legfontosabb RFC dokumentumok sorszámait az 1.2. Táblázatban foglaltuk össze.

1.2. Táblázat *Az internettel kapcsolatos több mint 2000 RFC közül a legfontosabbak*

Sorszám	Cím
791	Internet Protocol (IP)
792	Internet Control Message Protocol (ICMP)
793	Transmission Control Protocol
959	File Transfer Protocol
968	Twas the Night Before Start-up
1180	Bevezetés a TCP/IP protokollba
1188	Szabványjavaslat az adatcsomagok FDDI hálózatokon való átvitelére
1597	Magánhálózatok címkiosztása (Address Allocation for Private Internets)
2097	A PPP NetBIOS Frames Control protokoll
3300	Az internettel kapcsolatos protokollok hivatalos szabványai (Internet Official Protocol Standards 2/24/97)
4831	Hálózat alapú lokalizált mozgás kezelése (Network-Based Localized Mobility Management)

Összefoglalás

Ebben az órában megtudhattuk, mik is azok a hálózatok, és miért van szükség protokollokra, ha kommunikálni szeretnénk. Kiderült, hogy a TCP/IP fejlesztését az Egyesült Államok Védelmi Minisztériuma kezdeményezte, amely létrehozta az ARPAnet nevű kísérleti hálózatit, alapvető célja pedig egy olyan protokoll kidolgozása volt, amely a legkülönbözőbb környezeti feltételek esetén is teljes mértékben decentralizált hálózati működést tesz lehetővé.

Megismerkedtünk a TCP/IP néhány fontos szolgáltatásával, mint például a logikai címzéssel, a névfeloldással és az alkalmazások támogatásával. Megtudtuk, mely szervezetek felügyelik a TCP/IP fejlődését, illetve az RFC-k azok a dokumentumok, melyeket ezek a szervezetek megvitatnak, és amelyek egyben a TCP/IP és az internet „hivatalos dokumentációját" képezik.

Kérdések és válaszok

K *Mi a különbség a protokollt leíró szabvány és a protokoll megvalósítása között?*

V A protokollt leíró szabvány csupán szabályok gyűjteménye a protokoll megvalóstása (implementation) ezzel szemben egy konkrét szoftverkomponens amely a szabványban leírt szabályok alkalmazásával nyújt lehetőséget a hálózati kommunikációra.

K *Miért akartak az ARPAnet tervezői decentralizált hálózatit építeni?*

V Ők alapvetően katonai célokra tervezték ezt a hálózatot, így nem akarták a működéséhez létfontosságú szolgáltatásokat egy helyre összpontosítani, mert az kiváló célpont lehetett volna az ellenség számára.

K *Miért volt a végpontokon történő adatellenőrzés az ARPAnet egyik fontos szolgáltatása?*

V Az ARPAnet-nek tervezéséből adódóan nem volt semmiféle központja, amely a működését vezérelte volna. Ennek megfelelően az üzenetet küldő és az azt fogadó számítógépnek magának kellett gondoskodnia az adatok ellenőrzéséről és a kommunikáció vezérléséről.

K *Miért használnak a nagy hálózatokban névfeloldást?*

V Az IP címekre nehéz visszaemlékezni, ellenben könnyű őket elgépelni. A DNS-stílusú tartománynevek ezzel szemben lehetővé teszik, hogy az IP címekhez neveket vagy értelmes szavakat társítsunk.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **ARPAnet** – Egy olyan kísérleti hálózat volt, amely történetileg a TCP/IP „szülőházának" tekinthető.
- **Tartománynév** (*Domain Name*) – Olyan alfanumerikus név, amit egy IP címhez rendelünk a TCP/IP DNS szolgáltatásán keresztül.
- **Átjáró** (*gateway*) – Olyan útválasztó, amely egy helyi hálózatot (LAN) kapcsol egy nagyobb hálózathoz. Gyakran ugyanezt a kifejezést használják az olyan speciális átválasztókra is, amelyek különböző protokollok vagy protokollváltozatok között végeznek átalakítást.
- **IP cím** (*IP address*) – Olyan logikai cím, amely alapján egy számítógépet vagy egyéb hálózati eszközt azonosíthatunk egy TCP/IP hálózatban.
- **Logikai cím** (*logical address*) – Olyan hálózati cím, amit a kommunikációs protokollt megvalósító szoftver segítségével rendelünk hozzá a gépekhez.
- **Névszolgáltatás** (*name service*) – Olyan szolgáltatás, amely az ember által olvasható, felhasználóbarát címeket hálózati címekké alakítja.
- **Fizikai cím** (*physical address*) – Olyan állandó cím, amit a gyárban égetnek be a hálózati csatolóba.
- **Port** (*kapu*) – Olyan belső cím, amely összeköttetést biztosít egy futó alkalmazás és a TCP/IP adatátviteli rétege között.
- **Protokollrendszer** – Szabványok és eljárások olyan gyűjteménye, amely leírja, miként kommunikáljanak egymással egy hálózat gépei.
- **RFC** (*Request for Comment*) – Olyan hivatalos műszaki leírás, amely a TCP/IP vagy az internet működésének valamely vonatkozását tárgyalja. Az RFC-k több helyről szabadon letölthetők, az elsődleges lelőhelyük azonban a *www.rfc-editor.org* webhely.
- **Útválasztó** (*router*) – Olyan hálózati eszköz, amely a logikai címek alapján továbbítja az adatforgalmat a megfelelő helyre és amellyel így egy nagyobb hálózat kisebb forgalmú alhálózatokra bontható.
- **TCP/IP** – Hálózati protokollcsomag melyet az interneten és számos egyéb hálózatban is használnak szerte a világon.

2. ÓRA

A TCP/IP működésének alapjai

Ebben az órában a következőkről lesz szó:

- A TCP/IP protokollrendszer
- Az OSI modell
- Adatcsomagok
- A TCP/IP protokollok kölcsönhatásai

A TCP/IP protokollok egy egész csomagja, maga a protokoll pedig – amint azt korábbról már tudjuk – szabályok és eljárások gyűjteménye. Az esetek túlnyomó többségében a TCP/IP alapú kommunikációval kapcsolatos szabályok betartásáról a hardver vagy a szoftver automatikusan gondoskodik, vagyis a felhasználónak egyáltalán nem kell foglalkoznia a részletekkel. Ugyanakkor a TCP/IP rendszer legalább alapszintű ismerete mégis elengedhetetlen akkor, ha mi magunk szeretnénk egy rendszert beállítani, vagy hibát keresünk egy hálózatban.

Ebben az órában áttekintjük a TCP/IP protokollrendszer főbb elemeit, illetve megvizsgáljuk, miként működnek ezek együtt az adatok küldése és fogadása során.

Az óra végére a következőkre fog fény derülni:

- A TCP/IP rendszer rétegei és az egyes rétegek szerepe.
- Az OSI protokollmodell rétegei valamint azok viszonya a TCP/IP rétegeihez.
- A TCP/IP protokollok által használt fejlécek felépítése, valamint szerepe a protokollverem (*protocol stack*) egyes rétegeiben.
- Az adatcsomag megnevezése a TCP/IP verem egyes rétegeiben.
- A TCP, az UDP és az IP protokoll, valamint ezek együttműködése a TCP/IP vermen belül.

A TCP/IP protokollrendszer

Mielőtt közelebbről is megvizsgálnánk a TCP/IP rendszer egyes elemeit, nem árt tisztázni, pontosan milyen feladatokat is kell ellátnia egy protokollrendszernek.

Egy protokollrendszer – mint amilyen a TCP/IP is – a következő alapvető funkciókért felelős:

- Az üzenetek felbontása olyan könnyen kezelhető darabokra, amelyek hatékony vihetők át az alkalmazott kommunikációs közegen.
- Kapcsolat teremtése a hálózati csatolóval.
- A címzés kezelése. A küldő számítógépnek képesnek kell lenni arra, hogy az adatokat a fogadó géphez irányítsa a megfelelő címzés által. A fogadó gépnek hasonlóan képesnek kell lennie arra, hogy felismerje a neki küldött üzenetet, vagyis hogy éppen fogadnia kell valamit.
- Az útválasztás kezelése. Az adatoknak el kell jutniuk a küldő gép alhálózatából a fogadó gép alhálózatába még akkor is, ha ezek történetesen egészen eltérő fizikai felépítéssel bírnak.
- Hibakezelés, az adatáramlás vezérlése és a visszaigazolás biztosítása. A megbízható kommunikációnak alapfeltétele, hogy a küldő és a fogadó fél egyaránt képes legyen felismerni a és korrigálni a hibás átvitelt, illetve szükség esetén szabályozni az adatforgalmat.
- Adatok fogadása egy alkalmazástól és azok átvitele a hálózaton.
- Adatok fogadása a hálózatból és azok továbbítása a megfelelő alkalmazáshoz.

A fent vázolt alapvető feladatok ellátására a TCP/IP tervezői kezdettől fogva egy moduláris rendszert képzeltek el. A TCP/IP protokollrendszer tehát olyan különálló komponensekre bontható, amelyek elvileg egymástól függetlenül is képesek működni. Minden egyes komponens a kommunikációs folyamat egy-egy mozzanatáért felelős.

Ennek a moduláris felépítésnek a legfőbb előnye az, hogy a különböző fejlesztőcégek képesek könnyen hozzáigazítani a protokollal kapcsolatos szoftvereiket a legkülönbözőbb hardverekhez és operációs rendszerekhez. Példának okáért kizárólag a hálózat-

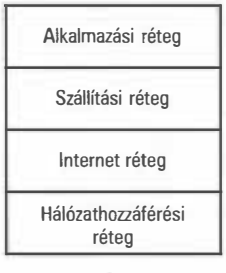
hozzáférési réteg (Network Access Layer; a 3. órában lesz róla szó bővebben) tartalmaz olyan szolgáltatásokat, amelyek a fizikai hálózat tervezésével és felépítésével kapcsolatosak. Ennek megfelelően egy gyártónak, például a Microsoftnak nem kell külön TCP/IP csomagot készítenie az optikai és a közönséges Ethernet hálózatokhoz. A moduláris felépítés ezt szükségtelenné teszi, hiszen a felsőbb rétegeket nem érinti az a tény, hogy az adatok konkrétan hogyan jutnak át a hálózati közegen. Kizárólag a hálózathozzáférési réteg az, amit újra kell írni.

A TCP/IP protokollrendszer tehát rétegekre oszlik, amelyek valamennyien egy-egy specifikus feladatcsoportot látnak el (lásd a 2.1. ábrát). Ez a modell, amit veremnek (stack) is szokás hívni a TCP/IP egészen kora fejlődési fázisában alakult ki, olyannyira, hogy TCP/IP modellnek is szokás hívni. A TCP/IP hivatalosan meghatározott rétegeit és azok funkcióit a következő listában soroljuk fel. Hasonlítsuk össze a funkciók felsorolását azzal a korábbi listával, amelyben egy protokollrendszer általános funkcióit soroltuk fel, és rögtön láthatjuk, miként oszlanak el a különböző típusú felelősségek az egyes rétegek között.

A 2.1. ábrán látható négyrétegű modell a legáltalánosabban elterjedt a TCP/IP működésének leírására. Ugyanakkor talán nem fölösleges megjegyezni, hogy nem ez az egyetlen modell. Az RFC 871-ben leírt ARPAnet modell például háromrétegű. Van benne egy hálózati interfész réteg (*Network Interface Layer*), egy gép-gép réteg (*Host-to-Host Layer*) és egy folyamatszintű/alkalmazási réteg (*Process-level/Application Layer*). Más leírások a TCP/IP rendszerét ötrétegűnek tekintik, ahol is a hálózathozzáférési réteg (*Network Access Layer*) az OSI modellel való párhuzam okán szétválik egy fizikai (*Physical Layer*) és egy adatkapcsolati rétegre (*Data Link Layer*). Megint más modellek egyszerűen kihagyják a a hálózathozzáférési vagy az alkalmazási réteget, mivel ezek nem annyira egységesek és viszonylag nehezebben körvonalazhatók, mint a közbenső rétegek.

Ami a rétegek elnevezését illeti, az szintén változó. Az ARPAnet rétegek nevei a mai napig felbukkannak a TCP/IP egyes leírásaiban, az internet réteget (*Internet Layer*) pedig néha hálózatközi rétegnek (*Internetwork Layer*), vagy egyszerűen hálózati rétegnek (*Network Layer*) hívják.

Ebben a könyvben végig a 2.1. ábrán bemutatott négyrétegű modellt fogjuk használni.



2.1. ábra
A TCP/IP modell protokollrétegei

- **Hálózathozzáférési réteg** (*Network Access Layer*) – Felületet biztosít a fizikai hálózathoz. Az átviteli közegnek megfelelően formálja meg a küldendő adatokat, a csomagokat pedig a fizikai címek alapján irányítja az alhálózat megfelelő eleméhez. Hibakezelést biztosít a fizikai hálózaton átvitt adatok ellenőrzéséhez.
- **Internet réteg** (*Internet Layer*) – Logikai, a hardvertől teljesen független címzést biztosít, ami lehetővé teszi, hogy az adatok fizikailag eltérő felépítésű alhálózatok között is átvihetők legyenek. Útválasztást biztosít a forgalom csökkentése végett és a támogatja a hálózatközi (internetwork) átvitelt. A „hálózatközi átvitel" (internetwork delivery) kifejezés jelen esetben helyi hálózatok (LAN) egy nagyobb, összekapcsolt rendszerét, illetve az ebben történő adatátvitelt takarja. Ilyen egy nagyobb vállalat belső hálózata, de tulajdonképpen maga az internet is. Végezetül ez a réteg teremti meg a kapcsolatot a logikai címek és a hálózathozzáférési réteg által használt fizikai címek között.
- **Szállítási réteg** (*Transport Layer*) – Folyamatszabályozási és visszaigazolása szolgáltatásokat, valamint hibakezelést biztosít a hálózatközi adatátvitelhez. Interfészként szolgál a hálózati alkalmazások működéséhez.
- **Alkalmazási réteg** (*Application Layer*) – Alkalmazásokat (segédeszközöket) biztosít a hálózati hibakereséshez, fájlok átviteléhez, távvezérléshez, valamint egyéb interneten végezhető tevékenységekhez. Tartalmaz ezen kívül egy vagy több alkalmazásprogramozói felületet (Application Programming Interface; API) is, amely egy adott operációs rendszerre írt egyéb programok számára lehetővé teszi a hálózati szolgáltatások elérését.

A későbbi fejezetekben természetesen részletesen foglalkozunk majd a TCP/IP verem valamennyi rétegének működésével. Amikor a TCP/IP protokollt megvalósító szoftver előkészül az adatátvitelre, a küldő oldalán a verem minden egyes rétege hozzátesz a küldendő adatokhoz valamennyi rá specifikus információt, amely a fogadó oldalán ugyanennek a rétegnek a működését fogja vezérelni. A küldő számítógép internet rétege például olyan információt csatol az adatcsomagokhoz, amely a túloldalon szintén az internet rétegnek lesz érdekes. Ezt a folyamatot szokás befoglalásnak (*encapsulation*) is hívni. A fogadó oldalon az egyed rétegek fokozatosan eltávolítják a nekik szóló információt az adatfolyamból, miközben az fölfelé halad a protokollveremben.

A protokollrendszerek szintjeit (mint amilyenek a 2.1. ábrán is láthatóak) az egész informatikai szakmában rétegekként említik. A rétegek az adatátvitel során fejlécinformációval (*header information*) látják el a rajtuk keresztülhaladó adatcsomagokat. (Erről a momentumról ebben a fejezetben még bővebben is esik majd szó.) amikor azonban magukról a konkrét komponensekről esik szó, a „réteg" kifejezés szükségszerűen kissé metaforikussá válik.

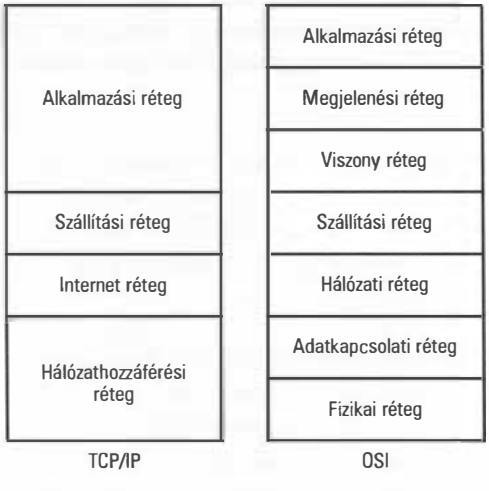
A 2.1. ábrához hasonló diagramok azt hivatottak szemléltetni, hogy az adatok számos interfészen haladnak keresztül útjuk során. Amíg ezeket a felületeket változatlan formában biztosítja egy rendszer, addig az egyes összetevőkön belüli történések részletei teljességgel lényegtelenek a többi komponens számára.

Ha a 2.1. ábrát kilencven fokkal elforgatjuk, teljesen olyan lesz, mint egy összeszerelő szalag, ami szintén gyakran használt analógia a protokollok komponenseivel kapcsolatban. Az adatok minden egyes választóvonalnál megállnak egy kicsit, de amíg az „anyag" minden ilyen ponthoz az előírásoknak megfelelő állapotban érkezik meg, addig a szalag komponensei egymástól függetlenül működhetnek.

A TCP/IP és az OSI modell

A hálózati kommunikációval foglalkozó ipar rendelkezik egy hétszintű hálózati modellel, amit OSI (Open Systems Interconnection) modellnek neveznek. Az OSI modell az ISO (International Standards Organization) dolgozta ki azzal a céllal, hogy szabványosítsa a hálózati protokollok tervezésének folyamatát, biztosítva ezzel az információhoz való nyílt hozzáférést illetve elősegítendő a különböző hálózati megoldások közti együttműködést.

A dologgal az egyetlen gond az, hogy a TCP/IP már régen haladt a maga útján, mire az OSI modell elkészült, így szigorú értelemben a TCP/IP modell nem felel meg az OSI modellnek. Ugyanakkor a TCP/IP konkrét megvalósításaira nagy hatással volt az OSI modell, ami ma leginkább abban látszik, hogy műszaki leírásokban gyakran láthatjuk, amint az OSI terminológiát a TCP/IP protokollveremre alkalmazzák.



2.2. ábra
A hétrétegű OSI modell

A 2.2. ábra a négyrétegű TCP/IP modell és a hétrétegű OSI modell közti megfeleltetést szemlélteti. Figyeljük meg, hogy az OSI modell az alkalmazási réteg (*Application Layer*) funkcióit három különálló réteg között osztja szét. Ezek az alkalmazási (*Application*), a megjelenési (*Presentation*) és a viszony (*Session*) rétegek. Hasonló figyelhető meg a hálózathozzáférési réteg (*Network Access Layer*) esetében is. Az OSI modell ezt két réteggel valósítja meg: az adatkapcsolati (*Data Link Layer*) és a fizikai (*Physical Layer*) ré-

tegekkel. Az OSI modellben a rétegek megnövekedett száma egyrészt növeli ugyan a rendszer összetettségét, másrészt azonban nagyobb rugalmasságot tesz lehetővé a megvalósítás terén. Ez különösen igaz a kettéválasztott hálózathozzáférési réteg esetében, ahol a így világosan különválnak a kommunikáció szervezésével és a fizikai közeghez való hozzáféréssel kapcsolatos funkciók. Az alkalmazási réteg esetében a hármas bontás szintén nagyobb rugalmasságot enged a fejlesztőknek abban, miként valósítják meg egy hálózati alkalmazás és a protokollverem közti kapcsolatot.

Az OSI modell a következő hét réteget különbözteti meg:

- **Fizikai réteg** (*Physical Layer*) – Az átvinni kívánt adatfolyamot olyan digitális vagy analóg villamos jelek sorozatává alakítja, amelyek ténylegesen áthaladnak az átviteli közegen. Az adatátvitel legalacsonyabb szintű felügyeletét is ez a réteg látja el.
- **Adatkapcsolati réteg** (*Data Link Layer*) – Interfészt biztosít a hálózati csatolóhoz. Logikai kapcsolatot tart fenn az alhálózattal.
- **Hálózati réteg** (*Network Layer*) – Biztosítja a logikai címzést és az útválasztást.
- **Szállítási réteg** (*Transport Layer*) – Hibakezelést és folyamatszabályozást biztosít a hálózatközi kommunikációban.
- **Viszony réteg** (*Session Layer*) – Munkafolyamatokat (session) létesít az egymással kommunikáló alkalmazások között.
- **Megjelenési réteg** (*Presentation Layer*) – Szabványos formára hozza az átvinni kívánt adatokat. Szükség esetén titkosítást és adattömörítést is végez.
- **Alkalmazási réteg** (*Apllication Layer*) – Az alkalmazások számára biztosít hálózati interfészt. Támogatja a hálózati programok közti fájlátvitelt, a kommunikációt és így tovább.

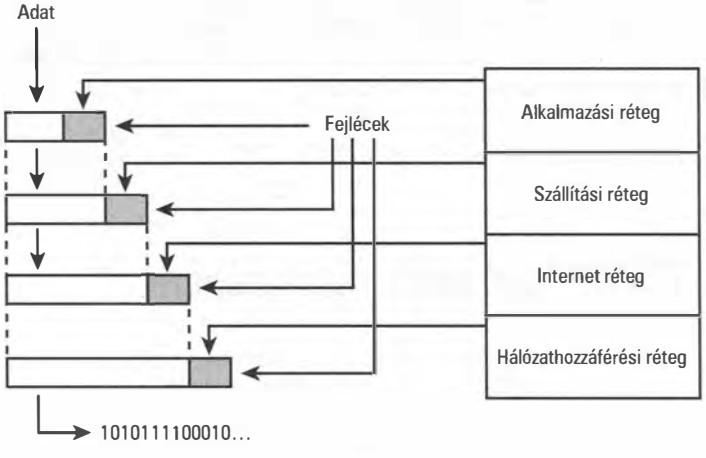
Szeretnék megint emlékeztetni arra, hogy a TCP/IP modell és az OSI modell csupán szabványok, nem megvalósítások. Ami azt illeti, a TCP/IP tényleges megvalósításai egyes esetekben kissé el is térhetnek a 2.1. és 2.2. ábrákon bemutatott elvi modelltől, sőt, a 2.2. ábrán bemutatott diagram egyes részletei a mai napig vita tárgyát képezik bizonyos szakmai körökben.

Figyeljük meg, hogy a TCP/IP és az OSI modell leginkább a szállítási (*Transport*) és az internet (az OSI modellben hálózati) rétegél hasonlít egymásra. Ezek a rétegek tartalmazzák a legkönnyebben azonosítható és megkülönböztethető elemeket és szolgáltatásokat, így egyáltalán nem véletlen, hogy a protokollrendszereket leggyakrabban a szállítási és a hálózati rétegükben található protokollok alapján nevezik el. Ami a TCP/IP protokollcsomagot illeti, a TCP a szállítási az IP pedig egy hálózati réteghez tartozó protokoll.

Adatcsomagok

A legfontosabb dolog, amire emlékeznünk kell a TCP/IP protokollverem működésével kapcsolatban az, hogy minden réteg játszik valamilyen szerepet a kommunikáció folyamatában. Működése közben minden réteg meghív bizonyos szolgáltatásokat, amelyek szükségesek ahhoz, hogy betölthesse a funkcióját. Amint egy kimenő adatcsomag halad lefelé a veremben, minden egyes réteg hozzátesz néhány lényeges információt, az úgynevezett **fejlécet** (*header*). Ez a fejléc aztán a tényleges adatokkal együtt utazik tovább a következő rétegbe, amely szintén hozzáilleszti az egészhez a saját fejlécét, és így tovább. A folyamatot a 2.3. ábra szemlélteti. Amikor az adatok megérkeznek rendeltetési helyükre, a folyamat visszafelé játszódik le. Amint a csomag halad fölfelé a veremben, minden egyes réteg leveszi róla a neki szóló fejlécet, és fölhasználja a benne tárolt információt.

Ami a veremben lefelé haladó csomagot illeti, a fejlécek fokozatos hozzáillesztése meglehetősen hasonlít azokra az orosz babákra, amelyeket egymásba lehet pakolni. Belül van a legkisebb baba, aztán körülötte az egyre nagyobbak sorakoznak. A fogadó oldalon a folyamat megfordul. A külső babák egyenként lekerülnek a csomagról, míg végül elérkezünk a legbelső babához. A fogadó oldal internet rétege azt az információt fogja fölhasználni, amit a küldő internet rétege helyezett el a megfelelő fejlécben. Hasonlóan a fogadó oldal szállítási rétege a küldő oldal szállítási rétege által elhelyezett információt olvassa ki. A csomagolás minden esetben tartalmazza azt az információt, amit a fogadó oldalmegfelelő rétegének tudnia kell az adatok helyes feldolgozásához. Mivel minden rétegnek más és más a funkciója, a feldolgozott adatcsomag is más-más alakot ölt.



2.3. ábra

Az adatokat minden réteg átcsomagolja úgy, hogy hozzájuk illeszti a saját fejlécinformációját.

A hálózati szakértők körében legalább annyi analógia kering a fent leírtakkal kapcsolatban, ahány rövidítést ismernek. Az imént említett orosz babás analógia csupán egy a sok közül, és mint bármely hasonlattól, ettől se kell túl sokat várni. Érdemes például megjegyezni, hogy az olyan hálózattípusok esetében, mint amilyen például az Ethernet, az adatok a hálózathozzáférési rétegben tovább darabolódnak kisebb egységekre. Ez babákra lefordítva valami olyasmit jelentene, hogy fogjuk az egymás körül koncentrikusan elhelyezkedő babákat, szépen fölszeleteljük őket, aztán a szeletekből kisebb babákat gyúrunk. A kisebb babák végül egyesekké és nullákká változnak és szépen elutaznak, ahova kell. A túloldalon az egyesek és nullák újra összeállnak előbb kis babákká, a kis babák babaszeletekké, a szeletek koncentrikusan egymásba helyezett nagy babákká, és a végén visszakapjuk azt, amiből elindultunk. Na, ezért van az, hogy sokan nem szeretik ezt az egyébként jobb sorsra érdemes babás metaforát.

Mint említettük, az adatcsomag minden egyes rétegben egy kicsit máshogy néz ki, sőt, kicsit máshogy is hívják attól függően, hogy éppen hol tartózkodik a veremben. Íme a lista:

- Az alkalmazási rétegben előállított adatcsomag az **üzenet** (*message*).
- A szállítási rétegben előálló adatcsomag, amely becsomagolva tartalmazza az alkalmazási réteg adatait a **szegmens** (*segment*) nevet viseli abban az esetben, ha a szállítási réteg TCP protokollja kezeli. Ha ellenben az UDP protokoll kapta meg feldolgozásra az adatokat, a csomag a **datagram**.
- Az internet rétegben előálló újabb csomagfajta, amely tehát a szállítási réteg adatainak becsomagolásával állt elő, megint csak a **datagram** nevet kapta a szaknyelvben.
- Végül a hálózathozzáférési rétegben az adatok megint újracsomagolódnak, illetve a datagramok szükség esetén tovább darabolódnak, az előálló csomagok pedig az **adatkeretek** (*frame*). A folyamat legvégén ezek az adatkeretek alakulnak bitfolyamokká és lépnek be ténylegesen az átviteli közegbe.

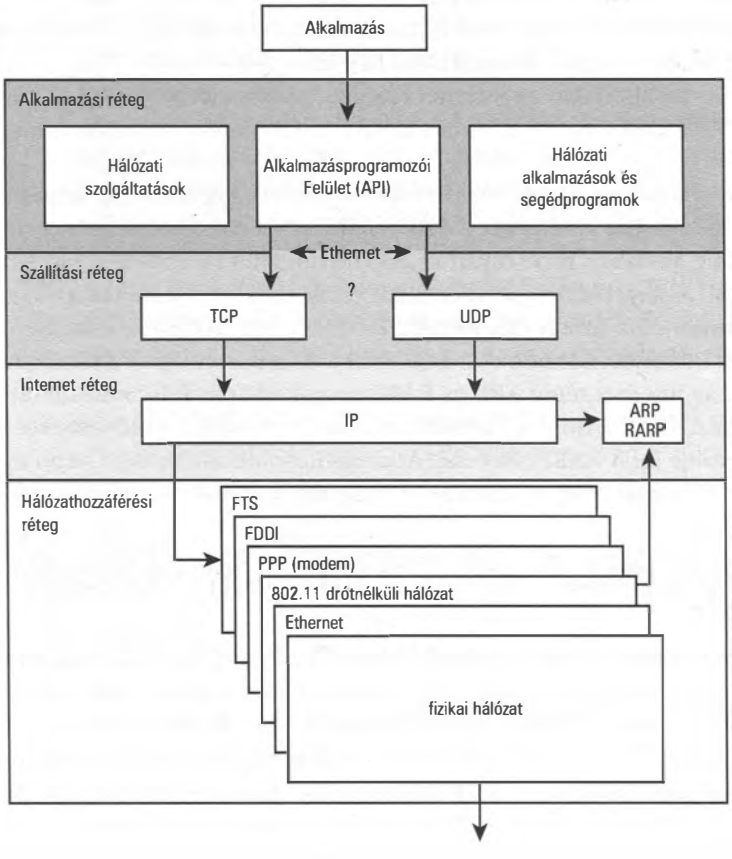
A későbbi fejezetekben minden egyes réteg működéséről és adatcsomagjairól részletesen lesz még szó.

Gyors pillantás a TCP/IP hálózatok működésére

Az a gyakorlat, hogy a protokollrendszereket az őket alkotó rétegek alapján írjuk le, csaknem teljesen szokványosnak nevezhető. A rétegződés olyasmi, ami tényleg betekintést enged egy protokollrendszer működésére, és ez olyannyira igaz, hogy a TCP/IP-t például gyakorlatilag lehetetlen leírni anélkül, hogy előtte be ne vezetnénk a réteges felépítést. Ugyanakkor ha kizárólag a rétegekre koncentrálunk, azzal elvesztünk bizonyos részleteket.

Először is ha a protokollrétegről beszélünk, nem pedig magukról a tényleges protokollokról, azzal az absztrakció egy újabb szintjét vezetjük be egy amúgy sem egyszerű rendszerben. Másodszor ha a konkrét protokollokról csak egy nagyobb egység, jelen esetben a protokollréteg részeként beszélünk, azzal azt a hamis látszatot keltjük, hogy minden protokoll egyformán fontos. Márpedig az igaz ugyan, hogy a TCP/IP protokollrendszer minden egyes protokolljának megvan a maga szerepe, de ami a rendszer működésének lényegét illeti, azt leírhatnánk akár úgy is, hogy csak néhány igazán fontos elemet említünk meg. Éppen ezért néha célszerű előrevenni a tárgyalásban ezeket a valóban fontos összetevőket, és átmenetileg a háttérben hagyni azt a bizonyos rétegzett hierarchiát, amiről egészen eddig szó volt.

A 2.4. ábra egy TCP/IP alapon működő hálózat leglényegesebb elemit mutatja. Természetesen egy hálózatban vannak más protokollok és szolgáltatások is, nem csak azok, amelyek itt láthatók, de a 2.4. ábra bemutatja mindannak a lényegét, ami egy hálózatban folyik.



2.4. ábra
Egy csupán az alapvető elemeket tartalmazó TCP/IP hálózat gyors áttekintése.

A helyzet alapszinten tehát a következő:

1. Az átvinni kívánt adatok származhatnak egy protokolltól, egy hálózati szolgáltatástól, vagy egy programozási felülettől (API), amely az alkalmazási rétegben működik. Ezek az adatok egy TCP vagy egy UDP kapunk (porton) keresztül juthatnak el a szállítási réteg (Transport Layer) két protokollja (TCP vagy UDP) közül az egyikhez. A programok tehát a hálózatot TCP vagy UDP protokollon át érhetik el attól függően, hogy mire van szükségük.
    - A TCP egy kapcsolatközpontú (connection-oriented) protokoll. Amint arról a 6. órában majd részletesen is lesz szó, a kapcsolatközpontú protokollok sokkal kifinomultabb folyamatvezérlést és hibakezelést biztosítanak, mint a kapcsolatmentes (connectionless) protokollok. Ez a gyakorlatban annyit jelent, hogy a TCP protokoll minden tőle telhetőt megtesz annak érdekében, hogy az adatok hibátlanul és teljes egészében érkezzenek meg a fogadóhoz. A TCP tehát megbízhatóbb, mint az UDP, viszont a kiterjedt hibaellenőrzés és folyamatszabályozás miatt egyben lassúbb is.
    - Az UDP kapcsolatmentes (connectionless) protokoll. Gyorsabb, mint a TCP, de egyáltalán nem annyira megbízható. Az UDP a hibaellenőrzést javarészt az őt használó alkalmazásra hagyja.
2. Az adatszegmens továbbhalad az internet rétegbe, ahol az IP protokoll ellátja a logikai címzéssel kapcsolatos információkkal és becsomagolva datagramot állít elő belőle.
3. Az IP datagram átkerül a hálózathozzáférési rétegbe, ahol olyan szoftverkomponensek veszik kezelésbe, amelyeket a fizikai hálózathoz való hozzáférésre terveztek. Ez a réteg általában több olyan kisebb adatkeretet (frame) állít elő az eddig datagramból, amelyek alkalmas méretűek és alkalmasan formáltak arra, hogy bekerülhessenek a fizikai hálózatba. Az olyan helyi hálózatok esetében, mint például az Ethernet hálózatok az adatkeret tartalmazhat olyan fizikai címeket is, amelyek az internet réteg ARP és RARP protokolljai által fönntartott táblázatokból származnak. Az ARP (Address Resolution Protocol) a az IP címeket fizikai címekké fordítja le. A RARP (Reverse Address Resolution Protocol) épp ennek az ellenkezőjét teszi, vagyis a fizikai cím alapján adja vissza a neki megfelelő IP címet.
4. Az adatkeret végül bitfolyammá alakul, ami a fizikai hálózaton keresztül továbbítódik a megfelelő helyre.

Természetesen végtelen azoknak az apró műszaki részleteknek a száma, amelyek leírják, hogy pontosan mit tesznek az egyes rétegek, miközben ellátják a fent tömören vázolt feladatukat. Egyelőre fogalmunk sincs például arról, hogyan szabályozza a TCP az adatáramlást, vagy hogyan tartja fönn az ARP és a RARP azokat a táblázatokat, amelyek alapján a fizikai címek IP címekké fordíthatók le és fordítva, vagy hogy honnan tudja az IP, hova kell küldenie azt a datagramot, amelynek címzettje egy másik alhálózatban található. Mindezekről természetesen részletesen lesz szó a későbbi fejezetekben.

Összefoglalás

Ebben az órában a TCP/IP protokollverem rétegeiről, és azok kapcsolatrendszeréről volt szó. Megtanultuk, milyen kapcsolatban áll egymással ez a négyrétegű szerkezet, és a hétrétegű OSI modell. Már tudjuk, hogy az adatcsomagokat minden egyes réteg becsomagolja némi kiegészítő információba, amely úgy van megfogalmazva, hogy az a fogadó oldal ugyanezen rétegének hasznos legyen. Megvizsgáltuk, milyen információkat tartalmaznak az egye rétegek által az adatcsomaghoz illesztett fejlécek, illetve milyen szakkifejezéssel illetjük az egyes rétegekben így előálló nagyobb csomagokat. Végezetül kizárólag a leglényegesebb protokollokat (TCP, UDP, IP, ARP, RARP) megemlítve vetettünk egy gyors pillantást egy a TCP/IP hálózat működésére.

Kérdések és válaszok

K *Mi a legalapvetőbb előnye a TCP/IP moduláris felépítésének?*

V A TCP/IP moduláris felépítésének köszönhetően a konkrét megvalósítás során a protokollverem könnyen hozzáigazíthat bármilyen hardverhez vagy operációs rendszerhez.

K *Milyen szolgáltatásokat nyújt a hálózathozzáférési réteg?*

V A hálózathozzáférési réteg szolgáltatásai alapvetően a fizikai hálózat igényeivel kapcsolatosak. Ezek a szolgáltatások teszik lehetővé az adatkeretek előkészítését, küldését és fogadását egy konkrét fizikai közegen, például egy Ethernet hálózaton át.

K *Az OSI modell melyik rétege feleltethető meg a TCP/IP verem internet rétegének?*

V A TCP/IP verem internet rétege az OSI modell hálózati rétegének felel meg.

K *Miért csatol a TCP/IP verem minden egyes rétege egy-egy fejlécet az adatcsomaghoz?*

V Mivel a fogadó oldal minden egyes rétegének más-más információra van szüksége a beérkezett adatok helyes feldolgozásához, ezért a küldő oldal rétegei az említett fejlécek formájában hozzáteszik ezeket az információkat az átküldött adatokhoz.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **Alkalmazási réteg** (*Application Layer*) – A TCP/IP verem azon rétege, amely a hálózati alkalmazások működését támogatja és interfészt nyújt a helyi operációs rendszernek.
- **Datagram** – Olyan adatcsomag, amely az internet rétegből a hálózathozzáférési rétegbe kerül, vagy olyan, amelyet az UDP protokoll továbbít a szállítási rétegbe.
- **Adatkeret** (*frame*) – A hálózathozzáférési réteg által előállított adatcsomag.

- **Fejléc** (*header*) – Olyan az adott protokollal kapcsolatos információcsomag, amit minden egyes réteg hozzáilleszt az általa feldolgozott adatcsomaghoz.
- **Internet réteg** (*Internet Layer*) – A TCP/IP csomag logikai címzést és útválasztást biztosító rétege.
- **IP** (*Internet Protocol*) – Az internet rétegnek az a konkrét protokollja, amely az útválasztást és a logikai címzést biztosítja.
- **Üzenet** (*message*) – A TCP/IP protokollcsaláddal kapcsolatos terminológiában az üzenet az az adatcsomag, amit az alkalmazási réteg továbbít a szállítási rétegbe. Maga a kifejezés ugyanakkor általánosan is használatos bármilyen hálózati kommunikáció leírása során. Ilyenkor bármilyen adatot jelent, ami a hálózat egyik tagjától a másikig jut el, vagyis nem minden összefüggésben kapcsolatos az alkalmazási réteggel.
- **Hálózathozzáférési réteg** (*Network Access Layer*) – A TCP/IP verem azon rétege, amely a fizikai hálózattal teremti meg a kapcsolatot.
- **Szegmens** (*segment*) – Olyan adatcsomag, amit a szállítási réteg TCP protokollja küld az internet rétegnek.
- **TCP** (*Transmission Control Protocol*) – A szállítási réteghez tartozó megbízható, kapcsolatközpontú (connection-oriented) protokoll.
- **Szállítási réteg** (*Transport Layer*) – A TCP/IP verem azon rétege, amely hibakezelést és visszaigazolási szolgáltatásokat nyújt, valamint interfészt képez a hálózat és a hálózati alkalmazások között.
- **UDP** (*User Datagram Protocol*) – A szállítási réteghez tartozó nem megbízhat, kapcsolat nélküli protokoll.

II. RÉSZ

A TCP/IP protokollrendszer

3. ÓRA

A hálózathozzáférési réteg

Ebben az órában a következőkről lesz szó:

- Fizikai címek
- Hálózati architektúrák
- Ethernet keretek

A hálózathozzáférési réteg (*Network Access Layer*) a TCPIP protokollverem legalsó rétege. Ez olyan szolgáltatásokat és specifikációkat tartalmaz, amelyek közvetlenül a hálózati hardverhez való hozzáférést, illetve az ezen a szintem zajló folyamatok szabályozását teszik lehetővé. Ebben az órában ennek a rétegnek a feladatairól lesz szó, valamint arról, hogyan viszonyulnak ezek az OSI modell megfelelő előírásaihoz. Szintén szó esik az összefoglaló néven Ethernetnek nevezett hálózati technológiáról. Az óra végére a következőkkel leszünk tisztában:

- Mik a hálózathozzáférési réteg feladatai?
- Hogyan viszonyul a TCP/IP hálózathozzáférési rétege az OSI modell megfelelő részéhez?
- Mik a feladatai egy hálózati architektúrának?
- Milyen részekből áll egy Ethernet keret?

Protokollok és a hardver

A hálózathozzáférési réteg a TCP/IP protokollverem talán legmisztikusabb és a különböző megvalósítások között a legkevésbé egységes rétege. Ez a réteg készíti elő és kezeli az adatokat úgy, hogy azok alkalmasak legyenek a fizikai hálózaton való átvitelre. Feladatai a következők:

- Interfészt biztosít a számítógép hálózati hardveréhez.
- Koordinálja az adatok átvitelét a kérdéses fizikai átviteli módszernek megfelelő konvenciók alapján.
- Olyan alakra hozza az adatokat, amely már közvetlenül átalakítható olyan digitális vagy analóg jelekké, amelyeket a kérdéses hálózattípus közvetíteni képes.
- Hibaellenőrzést végez a bejövő adatokon.
- Olyan hibaellenőrzési információt fűz hozzá a kimenő adatokhoz, amely alapján a fogadó ellenőrizni tudja azok sértetlenségét.

Természetesen minden a hálózathozzáférési réteg által elvégzett műveletet visszafelé is végre kell hajtani a fogadó oldalon, amint az adatok elérték a címzettet. A hálózathozzáférési réteg definiálja és vezérli azokat a folyamatokat, amelyek eredményeképpen az átvinni kívánt adatok elérik magát a hálózati hardvert, illetve az átviteli közeget. Ami a TCP/IP hálózathozzáférési rétege alatti dolgokat illeti, nos ott a hardver, a szoftver és az átviteli közeg igen összetett specifikációit, illetve ezek bonyolult összjátékát találjuk. Ezek általános tárgyalását sajnos teljességgel lehetetlenné teszi az a tény, hogy számtalan fizikai hálózattípus, illetve átviteli közeg létezik, amelyeknek mind megvannak a sajátos konvencióik. A hálózathozzáférési rétegnek minden konkrét esetben ezekhez a konvenciókhoz kell igazodnia, vagyis minden rendszer más és más lehet, annak konkrét fizikai felépítésétől függően.

A jó hír az, hogy a hálózathozzáférési réteg gyakorlatilag végig rejtve marad a közönséges felhasználó számára. A hálózati adapterhez mellékelt meghajtó, valamint az operációs rendszer megfelelő alacsony szintű elemei gyakorlatilag teljesen ellátják mindazokat a feladatokat, amelyek a TCP/IP modellben a hálózathozzáférési rétegre hárulnak. A felhasználónak általában csak néhány egyszerű beállítást kell megadnia a telepítés során, a továbbiakban a rendszer magától működik. Ráadásul a modern operációs rendszerek automatikus hardverfelismerési funkcióinak (*Plug and Play*) köszönhetően sok esetben egyáltalán semmiféle felhasználói beavatkozásra nincs szükség a felhasználó részéről a hálózati hardver telepítése során.

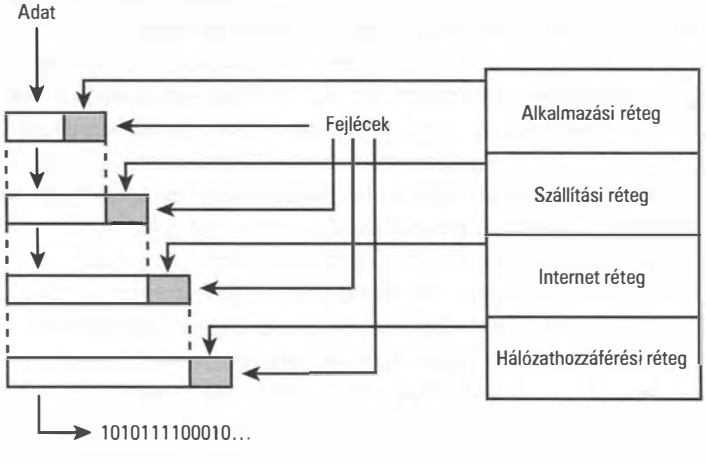
Mielőtt továbbhaladnánk, érdemes ismét kihangsúlyozni, hogy az 1., 2. , 4. és 5. órában tárgyalt IP címek logikai címek, vagyis kizárólag a szoftver szintjén léteznek. Ahhoz, hogy a protokollrendszer valóban képes legyen eljuttatni az adatokat a küldőtől a fogadóhoz, további, az adott LAN rendszerre specifikus címzési információra van szüksége, amelynek megszerzése és kezelése a hálózathozzáférési réteg feladata.

Talán érdemes megemlíteni, hogy a hálózathozzáférési réteg sokfélesége, összetettsége és láthatatlan volta számos szerzőt arra indított, hogy ennek a rétegnek a tárgyalását teljesen kihagyja a TCP/IP bemutatásából. Ezekben a könyvekben a közvetlenül a hardverrel kapcsolatban álló dolgokról mindössze annyit említenek, hogy a TCP/IP verem az internet réteg alatt található hálózati meghajtókra támaszkodik. A helyzet azonban az, hogy a TCP/IP szabvány szerint a hálózathozzáférési réteg magának a veremnek a része, vagyis egyetlen hálózati kommunikációval foglalkozó leírás sem lehet teljes ennek a legalább érintőleges tárgyalása nélkül.

A hálózathozzáférési réteg és az OSI modell

Amint azt már a 2. órában is említettük, a TCP/IP hivatalosan tulajdonképpen független a hétrétegű OSI modelltől. Ennek ellenére hálózati protokollrendszerek leírása során afféle általános keretrendszerként gyakran használják az OSI modellt, így most mi is ezt fogjuk tenni. Ez már csak azért is indokolt, mert az OSI terminológiája és alapkoncepciói különösen alkalmasak a hálózathozzáférési réteg funkcióinak tárgyalására, hiszen az OSI modell ezt a funkciókört további alrétegekre bontja. A részekre bontás pedig különösen hasznos, ha egy olyan összetett rendszer belső működését akarjuk feltérképezni, mint amilyen a hálózathozzáférési réteg.

Amint azt a 3.1. ábra is mutatja, a TCP/IP hálózathozzáférési rétege durván az OSI modell fizikai és adatkapcsolati rétegének felel meg. Az OSI modell szerint a fizikai réteg feladata az átvinni kívánt adatok olyan adatkeretekké tördelése, amelyek már alkalmasak arra, hogy az adott közegen átvihető bitfolyammá alakítsuk őket. Másként fogalmazva az OSI fizikai rétege az az összetevő, amely a közegben terjedő elektromos vagy analóg impulzusok kezelését és szinkronizálását végzi. A fogadó oldalon ugyanez a fizikai réteg rakja össze az impulzusokból az adatkereteket.



3.1. ábra
Az OSI modell és a hálózathozzáférési réteg kapcsolata

Az OSI modell szerint az adatkapcsolati rétegnek (Data Link Layer) alapvetően két funkciója van, így további két alrétegre bomlik:

- **Közeghozzáférést szabályozó alréteg** (*Media Access Control; MAC*) – Ez az alréteg biztosítja az interfészt a hálózati adapter felé. Ezért van az, hogy a hálózati meghajtó szoftverét gyakran nevezik (angolul) MAC meghajtónak (MAC driver), a hardvercímet pedig, amit a gyártó éget be magába az eszközbe szintén MAC cím (MAC address) néven említik a szakirodalomban.
- **Logikai kapcsolatot szabályozó alréteg** (*Logical Link Control; LLC*) – Ez az alréteg végzi az alhálózaton keresztül továbbított adatok hibaellenérzését, illetve logikai kapcsolatot tart fönn az alhálózat egymással kommunikáló tagjai között.

Ami a valós protokollmegvalósításokat illeti, a TCP/IP és az OSI modell rétegeinek egymás közti megfeleltethetőségét tovább árnyalja két fejlesztés. Az egyik az NDIS (Network Driver Interface Specification), amit a Microsoft és a 3Com Corp. fejlesztett, a másik az ODI (Open Data-Link Interface), amely az Apple és a Novell fejlesztése. Ezeket a szoftveres rétegeket azért hozták létre, hogy egy tetszőleges protkollverem (például a TCP/IP) egyszerre több hálózati adapter is használhasson, illetve hogy egy hálózati csatoló fölött több hálózati protokollverem is működhessen. Ez a két közbülső réteg gyakorlatilag lehetővé teszi, hogy a felsőbb rétegekhez tartozó protokollok függetlenül lebegjenek a hálózathozzáférési réteg fölött. Ez egyrészről újabb értékes funkciókkal gazdagítja a hálózati operációs rendszereket, másrészről viszont újabb összetettséget visz a szoftverkomponensek rétegzettségébe és azoknak a módszereknek a szisztematikus tárgyalásába, amelyek révén a felsőbb rétegek az alsóbbakkal tartják a kapcsolatot.

A hálózati architektúra

A helyi hálózatokat (LAN) a gyakorlatban nem is annyira a protokollrétegek mint inkább azok architektúrája alapján szokás vizsgálni. A hálózati architektúrát (*network architecture*) gyakran nevezik hálózattípusnak (*LAN type*) vagy hálózati topológiának (*LAN topology*) is. A hálózati architektúra – mint például az Ethernet – nem egyéb, mint olyan specifikációk összessége, amelyek a közeghez való hozzáférést, a fizikai címzést, a kommunikáló felek egymás közti viszonyát, illetve az átviteli közeghez való hozzáférés módját írják le. Amikor hálózati architektúrát választunk tulajdonképpen arról döntünk, milyen hálózathozzáférési réteget kívánunk használni hálózatunkban.

A hálózati architektúra tehát nem egyéb, mint a fizikai hálózat egyfajta terve, illetve azon specifikációk összessége, amelyek a fizikai hálózatban bonyolódó kommunikációt írják le. Mivel a kommunikáció mozzanatai szükségszerűen összefüggnek a hálózat fizikai adottságaival, az ezzel kapcsolatos specifikációk általában teljes csomagokat alkotnak. E csomagok a következő kérdésekkel kapcsolatos megfontolásokat tartalmaznak:

- **Hozzáférési módszer** (*access method*) – A hozzáférési módszer olyan szabályok gyűjteménye, amelyek azt határozzák meg, hogy a hálózatba kapcsolat gépek miként osztoznak meg az átviteli közegen. Az ütközések elkerülése végett a számítógépeknek ezeket a szabályokat kell követniük, mielőtt konkrét adatátvitelbe kezdenének.
- **Az adatkeretek (data frame) formátuma** – Az internet rétegből származó adatcsomagok nem változatlan formában kerülnek át a címzetthez, hanem adott formátummal bíró adatkeretekké alakulnak. Az adatkeretek fejlécének tartalmaznia kell azt az információt, amely alapján az adatkeret a fizikai hálózatban a megfelelő helyre továbbítható. Az adatkeretekről ebben az órában még részletesebben is esik szó.
- **A kábelezés típusa** – A hálózat kiépítéséhez használt kábelek tulajdonságai határozzák meg a bennük elektromos jelek formájában továbbítható bitfolyam egyes tulajdonságait, illetve a hálózati illesztő bizonyos jellemzőit.
- **A kábelezéssel kapcsolatos szabályok** – A használt protokollok, a kábelek típusa, valamint a vezetékekben továbbított villamos jelek jellemzői együttesen meghatároznak bizonyos fizikai korlátokat (például maximális vagy minimális kábelhossz), amelyeket a hálózat kiépítése során be kell tartani.

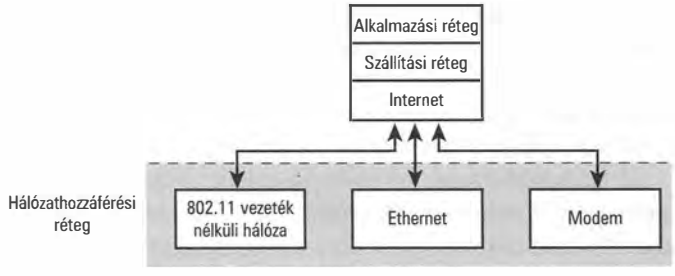
Az olyan részletek, mint a kábel vagy a csatlakozó típusa tulajdonképpen nem képezik a hálózathozzáférési réteg részét, ugyanakkor a szoftverkomponensek fejlesztőinek rendelkezniük kell bizonyos alapfeltevésekkel a fizikai réteg működésével kapcsolatban. Összességében tehát elmondható, hogy minden hardvertípushoz más és más meghajtószoftver használható.

Az egész rendszer lényege ugyanakkor az, hogy a felsőbb rétegeknek nem kell törődniük a hálózathozzáférési réteg konkrét felépítésével és belső működésével. A TCP/IP protokollverem rétegzett felépítése lehetővé teszi a hardver és a szoftver ilyen mértékű függetlenítését, hiszen minden, ami a hardverrel közvetlenül kapcsolatos, a hálózathozzáférési réteg dolga. Éppen ez az a tervezési megfontolás, ami miatt a TCP/IP olyan sok hardvertípuson illetve átviteli közeggel képes működni.

A következőkben felsorolunk néhány olyan architektúrát, amelyeket a TCP/IP hálózathozzáférési rétege képes támogatni:

- **IEEE 802.3 (Ethernet)** – Ez az a bizonyos kábelezett hálózat, amit a legtöbb irodában használnak.
- **IEEE 802.11 (drótnélküli hálózatok ; wireless networking)** – A köznapi életben – irodákban, lakásokban, kávéházakban – használt közönséges vezeték nélküli hálózat.
- **IEEE 802.16 (WiMAX)** – Olyan technológia, melyet nagy távolságokat átívelő, mobil, vezeték nélküli kapcsolatok kiépítésére használnak.
- **PPP (Point to Point Protocol)** – Telefonvonalon, modem segítségével megvalósított kapcsolattípus.

A fentieken kívül még számos egyéb olyan hálózati architektúra létezik, amelyet támogat a TCP/IP szabvány. Ugyanakkor amint az a 3.2. ábrán is látható, a lényeg valamennyinél az, hogy a hardverrel közvetlenül kapcsolatban álló komponensek önálló réteget alkotnak, amelyektől a felsőbb rétegek logikailag függetlenek abban az értelemben, hogy nem kell törődniük a hardverhez kapcsolódó szolgáltatások mikéntjével. A felsőbb rétegekhez kapcsolódó szolgáltatások, mint például a logikai címzés teljesen hardverfüggetlen módon valósíthatók meg.



3.2. ábra

Mivel a hálózathozzáférési réteg elrejti a külvilág elől a hálózati hardver működésének részleteit, a verem felsőbb rétegei gyakorlatilag hardverfüggetlen módon működhetnek.

3.3. ábra

A legtöbb hálózati operációs rendszer lehetővé teszi, hogy több különböző architektúrát rendeljünk a TCP/IP veremhez.

Bár a protokollrétegek egymás közti kommunikációja az egyszerű felhasználó előtt rejtve marad, azért a hardverrel kapcsolatos és az attól független rétegek kommunikációjának egyes részleteivel néha szembesülünk. Ilyen például az az eset, amikor az operációs rendszer megfelelő helyén meg kell adnunk a szokásos hálózati beállításokat. A 3.3. ábra például egy MacOS X rendszer hálózati konfigurációs ablakát mutatja, ahol egyszerre több különböző hálózati architektúrát rendelhetünk hozzá a TCP/IP verem szolgáltatásaihoz. A választható opciók között szerepel a közönséges Ethernet hálózat, a Bluetooth kapcsolat, a modem illetve az AirPort. Ez utóbbi egyébként tulajdonképpen nem más, mint az IEEE 802.11 vezeték nélküli LAN specifikáció Apple által kiegészített változata más néven újracsomagolva.

A modemekről, vezeték nélküli kapcsolatokról és egyéb hálózati technológiákról az elkövetkező órákban még részletesen lesz szó. Annak szemléltetésére azonban, hogy a hálózathozzáférési réteg konkrét megvalósításakor a programozóknak milyen problémákat kell megoldaniuk, a következő szakaszokban részletesen megvizsgáljuk az egyik legelterjedtebb hálózati architektúrát, az Ethernet hálózatot.

Fizikai címzés

Amint arról a korábbi fejezetekben már volt szó, a logikai IP címek és a hálózati adapterbe égetett egyedi fizikai címek összerendelését a hálózathozzáférési réteg végzi. A fizikai címet gyakran MAC címnek (MAC address) is nevezik, mivel az OSI modellben a fizikai címzés kezelése a közeghozzáférést vezérlő alréteg (Media Access Control ; MAC) feladata. Mivel a fizikai címzés részleteit a hálózathozzáférési réteg teljesen elrejti a külvilág elől, maguk a fizikai címek többféle formát is fölvehetnek attól függően, hogy milyen fizikai hálózatról van szó, és annak a specifikációja mit ír elő ezzel kapcsolatban.

Az Ethernek hálózatok esetében a fizikai címet a gyártó égeti be a hálózati adapterbe. Néhány évvel ezelőtt az ilyen hálózati csatolók gyakorlatilag kizárólag külön bővítőkártyák formájában léteztek, amelyeket az alaplap megfelelő csatlakozójában kellett dugni. Később megjelentek az olyan alaplapok, amelyek már integrálva tartalmazták az Ethernet vezérlőt. Akármilyen Ethernet eszközünk van is azonban, abban biztosak lehetünk, hogy az rendelkezik egy globálisan egyedi fizikai címmel, amit a gyártó rendelt hozzá.

A helyi hálózaton továbbított adatkeretek mindegyikének tartalmaznia kell a küldő és a fogadó fizikai címét, ez teszi lehetővé a gépek egyértelmű azonosítását. A hosszú, 48 bitből álló Ethernet címek ugyanakkor olyan mértékig nem felelnek meg az emberi igényeknek, hogy mi felhasználók, nem is ezeket használjuk a mindennapi életben. Az első ötlet ezzel a problémával kapcsolatban az lehetne, hogy kódolják át a hálózati verem felsőbb rétegeiben a „csúnya" címeket „szép" címekké. Ez az út azért nem járható, mert ezzel elveszítenénk a felsőbb rétegek hardverfüggetlenségét, hiszen a fizikai

cím az architektúrától függően többféle lehet, ami többféle átkódolást igényelne. Ehelyett a TCP/IP verem az emberileg fogyasztható IP címeket használja, amelyeket az ARP (*Address Resolution Protocol*) és a RARP (*Reverse Address Resolution Protocol*) protokoll fordít le fizikai címmé és vissza. Az ARP és a RARP tehát nem más, mint kétirányú kapocs a felhasználó által beállítható IP cím, és a felhasználó számára gyakorlatilag láthatatlan fizikai cím között. E két protokoll működésének részleteiről a 4. órában lesz szó az internet réteg kapcsán.

A következő szakaszokban az Ethernet hálózatok kapcsán említett „cím" tehát soha nem azonos az általunk ismert, az adapterhez logikailag hozzárendelt IP címmel, de egyértelműen leképeződik egy IP címmé az internet rétegben.

Az Ethernet

Az Ethernet kétségkívül a legnépszerűbb hálózati technológia a ma használatos kommunikációs megoldások közül. Népszerűségét bizonyára nem kis mértékben köszönheti elfogadható árának, valamint annak, hogy a kábelek, csatlakozók, és egyéb hálózati eszközök szintén könnyen és olcsón beszerezhetők, és könnyen felszerelhetők. Szinte biztosra vehető, hogy ha az olvasó valaha is benézett a számítógépe mögé, bizonyára látott már Ethernet csatlakozót és kábelt. Manapság ugyan egyre terjednek a vezeték nélküli hálózati megoldások, ez azonban egyelőre nem szorította háttérbe a vezetékes megoldást. A drótnélküli megoldások egyikét gyakran nevezik vezeték nélküli Ethernetnek (Wireless Ethernet) is, mivel az eredeti Ethernet specifikáció számos elemet tartalmazza.

Egy klasszikus Ethernet hálózatban a számítógépek egyetlen átviteli közegen osztoznak. Az Ethernet egy CSMA/CD-nek nevezett (*Carries Sense Multiple Access with Collision Detection; Vivőérzékelésen alapuló többszörös hozzáférés ütközésérzékeléssel*) módszert használ annak meghatározására, hogy egy számítógép mikor kezdhet sugározni a közegen keresztül. A CSMA/CD lényege, hogy a számítógépek folyamatosan figyelik a közeg foglaltságát, és várnak, amíg az szabaddá nem válik. Ha két gép mégis egyszerre kezdene sugározni, ütközés (*collision*) keletkezik. A két gép ekkor megáll, mindketten várakoznak egy véletlenszerűen megválasztott ideig, majd újra próbálkoznak.

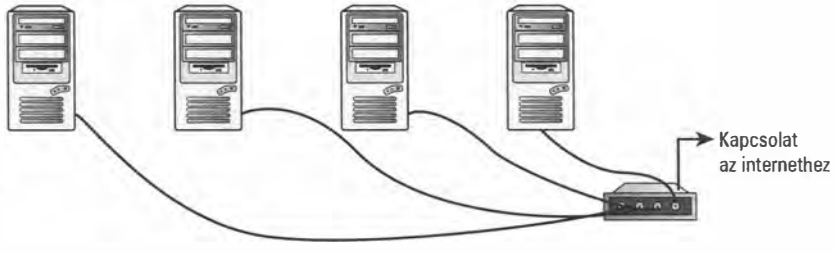
A CSMA/CD tehát működését tekintve olyan, mint az a protokoll, amit egy teremnyi udvarias ember követ, ha beszélgetni akarnak. Aki éppen szólásra szeretne emelkedni, előbb fülel, hogy jelenleg beszél-e valaki más. Ez a betűszó CS része (*Carrier Sense*). Ha véletlenül mégis ketten egyszerre kezdenének beszélni, mindketten érzékelik a problémát, elhallgatnak, várnak valameddig, majd újra beszélni kezdenek. Ez az ütközésérzékelés (CD).

A közönséges Ethernet kiválóan működik kis és közepes terhelés esetén, nagyobb terhelésnél azonban komoly problémát jelentenek az ütközések. Éppen ezért a modern Ethernet hálózatok már általában tartalmaznak olyan eszközöket (például kapcsolókat), amelyek speciális módon kezelik a kapcsolatokat és csökkentik az ütközések számát. A HUB-okról és kapcsolókról (switch) a 9. órában esik részletesebben szó.

Az Ethernet többféle átviteli közeggel képes elboldogulni. A hagyományos, HUB-okon alapuló 10BASE-T Ethernetet eredetileg 10 Mbps sebességre tervezték, manapság azonban már a 100 Mbps sebességgel működő úgynevezett gyors Ethernet található meg mindenütt. Szintén egyre gyakoribbak az 1000 Mbps sebességet biztosító Gigabit Ethernet megoldások. A korai Ethernet változatok kizárólag folytonos koaxiális kábeleket használtak átviteli közegként (3.4. ábra). Manapság a leggyakoribb megoldás ezzel szemben az, hogy valamennyi gép egy központi hálózati eszközhöz csatlakozik (3.5. ábra).

3.4. ábra

Az Ethernet korai változatainál valamennyi számítógép egyetlen közös koaxiális kábelre csatlakozott.

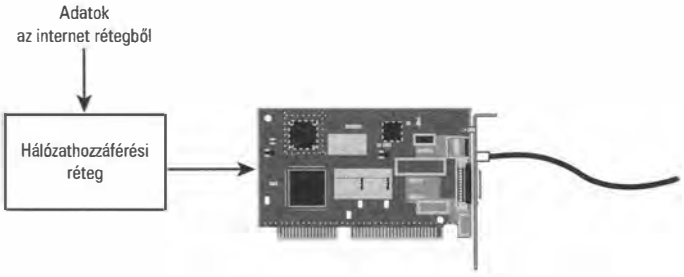


3.5. ábra

A modern Ethernet hálózatokban a számítógépek általában egy központi hálózati eszközhöz, például egy kapcsolóhoz (switch) csatlakoznak.

Egy Ethernet adatkeret anatómiája

A hálózathozzáférési réteg az internet rétegtől fogadja az adatcsomagokat, majd átalakítja azokat olyan formára, amely megfelel a fizikai hálózat specifikációinak (lásd a 3.6. ábrát). Az Ethernet hálózatok esetében ez konkrétan azt jelenti, hogy a hálózathozzáférési réteg az adatcsomagokat a gépben található hálózati kártya speciális igényeihez alakítja.



3.6. ábra

A hálózathozzáférési réteg a fizikai hálózat igényeinek megfelelően formája át a kapott adatokat.

Amikor az Ethernet kártya meghajtószoftvere adatokat kap az internet rétegtől, a következő műveleteket hajtja végre:

1. Szükség esetén olyan kisebb darabokra bontja az adatcsomag tartalmát, amelyek elférnek egy Ethernet keret adatmezőjében. Egy Ethernet keret teljes mérete legalább 64 de legfeljebb 1518 bájt lehet, amibe nem tartozik bele az úgynevezett preambulum (*preamble*). Egyes rendszerek ennél sokkal nagyobb, akár 9000 bájtos adatkereteket is támogatnak. Ezek az úgynevezett Jumbo keretetek (*Jumbo frames*) növelik a rendszer hatékonyságát, ugyanakkor fölvetnek bizonyos kompatibilitási problémákat is, így támogatottságok korántsem nevezhető általánosnak.
2. A földarabolt adatokat adatkeretekbe (*frames*) csomagolja. Minden ilyen keret tartalmaz bizonyos mennyiségű szállítandó adatot, illetve egyéb olyan információkat, amelyekre a hálózati kártyának van szüksége ahhoz, hogy megfelelően tudja továbbítani az üzeneteket. Egy az IEEE 802.3 szabványnak megfelelő Ethernet adatkeret a következőket tartalmazza:

   Preambulum (preamble) – Olyan bitsorozat, amely az adatkeret elejét hivatott jelezni. Hossza összesen 8 bájt, amelyből legalább egy bájt a határoló szerepét tölti be (*Start Frame Delimiter*).

   A fogadó címe – Annak a hálózati adapternek a 6 bájtos (48 bites) fizikai címe, amelynek fogadnia kell az adatokat.

   A forrás címe – Annak a hálózati adapternek a 6 bájtos (48 bites) fizikai címe, amely küldi az adott Ethernet keretet.

   Adathossz – 2 bájt (16 bit), amely az adatmező hosszát tartalmazza.

   Adatok – Az adatkeret hasznos tartalma, vagyis az átvinni kívánt adatmennyiség.

   FCS (Frame Check Sequence) – Az adatkeret 4 bájtos (32 bites) ellenőrző összege. Az FCS meglehetősen elterjedt módszer az adatátvitel hibátlan voltának ellenőrzésére. A módszer lényege, hogy a küldő kiszámít egy úgynevezett CRC (*Cyclical Redundancy Check*) értéket, és elhelyezi azt

az általa elküldött adatkeretben. A fogadó fél a kapott adatok alapján szintén kiszámítja ezt az értéket, és összehasonlítja a küldő által a csomag FCS mezőjébe tett számmal. Ha a kettő egyezik, az átvitel sikeres volt. Ha az összehasonlítás sikertelen, akkor biztosra vehető, hogy az adatkeret az átvitel során megsérült, és újra át kell küldeni.

3. Átadja az adatokat az alsóbb szinten levő komponenseknek, amelyek az OSI modell fizikai rétegének felelnek meg. A fizikai réteg a kerete bitfolyammá alakítja, majd átküldi azt az átviteli közegen.

A hálózat többi csatolókártyája megkapja a kiküldött adatkeretet, és megvizsgálja, hogy a benne található fizikai cím azonos-e a sajátjával. Ha igen, akkor az üzenet neki szól, a kártya az adatokat a protokollverem felsőbb rétegeinek továbbítja feldolgozásra.

Összefoglalás

Ebben az órában a hálózathozzáférési réteggel ismerkedtünk meg, amely vitán felül a TCP/IP protokollrendszer legösszetettebb, feladatait tekintve pedig a legszerteágazóbb funkciókkal rendelkező rétege. A hálózathozzáférési réteg alapvetően azokat a definíciókat és eljárásokat foglalja magában, amelyek a hálózati hardverhez illetve az átviteli közeghez való hozzáférést szabályozzák. Mint megtudtuk, számos különféle LAN architektúra létezik, és ennek megfelelően a hálózathozzáférési réteg felépítése is többféle lehet. Ebben az órában a lehetséges megvalósítások közül egyet közelebbről is megvizsgáltunk.

Az itt tárgyalt Ethernet technológia manapság meglehetősen elterjedtnek tekinthető, ugyanakkor fontos hangsúlyozni, hogy a hálózati kapcsolatok kiépítésének számos egyéb módja is létezik. Mindazonáltal minden hálózati technológiának rendelkeznie kell olyan eljárásokkal, amelyek felkészítik az adatokat a fizikai közegen való átvitelre, vagyis a TCP/IP protkollveremnek minden rendszeren kell hogy legyen hálózathozzáférési rétege. Az olyan egyéb hálózati technológiákról, mint a modemek, a vezeték nélküli hálózatok vagy a WAN-ok a következő órákban még esik szó.

Kérdések és válaszok

K *Milyen típusú szolgáltatások definícióit tartalmazz a a hálózathozzáférési réteg?*

V A hálózathozzáférési réteg alapvetően a fizikai hálózathoz való hozzáféréssel kapcsolatos eljárások specifikációit foglalja magában.

K *Mely OSI rétegek felelnek meg a TCP/IP hálózathozzáférési rétegének?*

V A hálózathozzáférési réteg nagyjából az OSI modell adatkapcsolati (*Data Link Layer*) és fizikai rétegének felel meg.

K *Melyik a legelterjedtebb LAN architektúra manapság?*

V Napjainkban az Ethernet tekinthető a legelterjedtebb hálózati technológiának, bár egyre nagyobb a drótnélküli hálózatok térnyerése.

K *Mi az a CSMA/CD?*

V A CSMA/CD (Carrier Sense Multiple Access with Collision Detect) egy az Ethernet technológia részét képező közeghozzáférési módszer. Lényege, hogy a forgalmazni kívánó számítógépek figyelik a közeg foglaltságát, megvárják, amíg senki más nem akar adatokat átvinni, és csak akkor kezdenek forgalmazni. Ha mégis ütközés lépne fel, vagyis két gép egyszerre kezd adni, akkor mindketten megállnak, véletlenszerűen választott ideig várnak, majd újra próbálkoznak.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- Hozzáférési módszer – Az átviteli közeghez való hozzáférést szabályozó módszer.
- CRC (*Cyclic Redundancy Check*) – Ellenőrzőösszeg, amely lehetővé teszi, hogy a fogadó ellenőrizhesse az adatkeret tartalmának helyességét.
- CSMA/CD – Az Ethernet technológia által használt közeghozzáférési módszer.
- Adatkeret (*data frame*) – Egy Ethernet hálózatban egyben továbbítható adatcsomag.
- Adatkapcsolati réteg (*Data Link Layer*) – Az OSI modell második rétege.
- Ethernet – Manapság meglehetősen népszerű LAN architektúra, amely a CSMA/CD közeghozzáférési módszert használja.
- LLC (*Logical Link Control sublayer*) – Az OSI modell adatkapcsolat rétegének (Data Link Layer) egyik alrétege. Feladata a hibaellenőrzés és a kommunikáló felek közti logikai kapcsolat kiépítése.
- MAC (*Media Access Control sublayer*) – Az OSI modell adatkapcsolati rétegének egyik alrétege. Feladata interfész biztosítás a hálózati hardver felé.
- Hálózati architektúra (*network architecture*) – A fizikai hálózat teljes specifikációja, amely magában foglalja a hozzáférési módszer, az adatkeretek és a kábelezés leírását.
- Fizikai cím (*physical address*) – Olyan állandó hálózati cím, amit a gyártó éget be a hálózati kártyába. A fizikai hálózatban történő kommunikáció során használatos.
- Fizikai réteg (*Physical Layer*) – Az OSI modell első rétege, amelynek feladata az adatkeret olyan bitfolyammá alakítása, amely már közvetlenül alkalmas a hálózaton való átvitelre.
- Preambulum (*preamble*) – Olyan bitsorozat, amely az átvitt adatkeret elejét hivatott jelezni.

4. ÓRA

Az internet réteg

Ebben az órában a következőkről lesz szó:

- IP címek
- Az IP fejléc
- ARP
- ICMP

Amint azt előző órában megtanultuk, egy helyi hálózat számítógépe egymással a hálózati csatoló fizikai címe alapján kommunikálnak. Ezt a szolgáltatást a hálózathozzáférési réteg biztosítja számukra. Ez így persze szép és jó, de maradt itt egy igen fontos kérdés, amire még nem kaptunk választ: hogyan jut el egy e-mail akár az egyik államból egy a másikban található címzetthez? És mindig pontosan! Erre a kérdésre ebben az órában fogjuk megkapni a választ, ami egyébként röviden annyi, hogy a helyi alhálózaton túli helyekre az internet réteg hivatott eljuttatni az adatokat. Ennek a rétegnek a három talán legfontosabb protokollja az IP, az ARP és az ICMP. Ebben az órában ezekről, és a következő kérdésekről esik szó:

- Mi a rendeltetése az IP, az ARP és az ICMP protokolloknak?
- Mi az a hálózati azonosító (*network ID*) illetve gépazonosító (*host ID*)?
- Mit nevezünk oktetnek (*octet*)?
- Hogyan alakul át egy pontokkal tagolt decimális cím annak bináris megfelelőjévé?
- Hogyan alakíthatunk egy 32 bites IP címet annak decimális megfelelőjévé?
- Miből áll egy IP fejléc (*IP header*)?
- Mi a rendeltetése egy IP címnek?

Címzés és kézbesítés

Amint azt a harmadik órában tisztáztuk, a számítógép egy hálózati interfész (hálózati kártya) segítségével kommunikál a többi géppel. Ennek az interfésznek van egy egyedi azonosítója, a hardvercím vagy fizikai cím, aminek a helyi hálózatban az a rendeltetése, hogy a különböző gépek ki tudják választani a forgalomból a nekik szóló csomagokat. A fizikai cím minden esetben egyedi, és a gyártó égeti be az eszközbe. Egy olyan hálózati eszköz, mint amilyen például egy Ethernet kártya, semmit nem tud a fölötte elhelyezkedő protokollrétegekről. Fogalma sincs, hogy mi az az IP cím, vagy hogy egy beérkező csomag a Telnet vagy az FTP programnak szól. A kártya csak annyit tesz, hogy figyeli a forgalmat, kiválogatja belőle azokat az adatkereteket, amelyekben a saját fizikai címét látja, és azokat továbbküldi a protokollverem felsőbb rétegeinek.

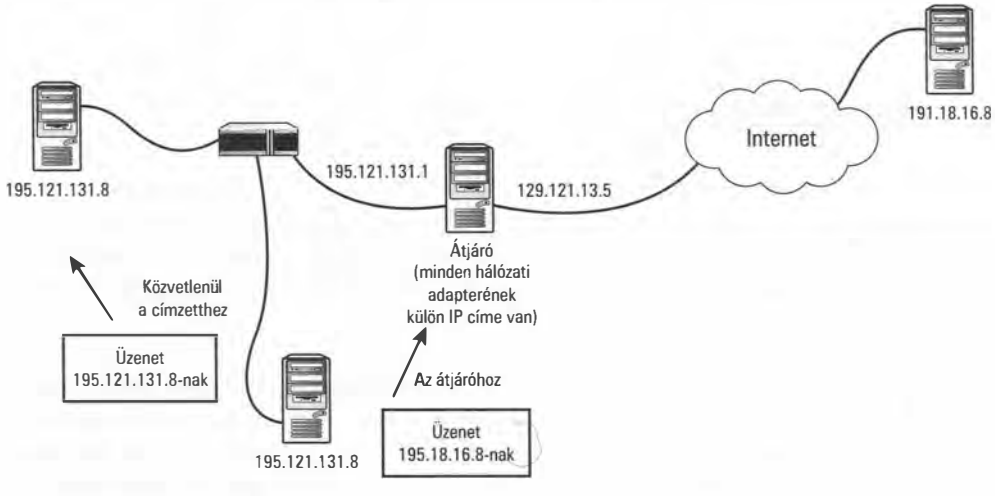
Ez a fizikai címzési séma tökéletesen működik az egyedi LAN szegmenseken. Igazából egy olyan hálózatban, amelynek csupán néhány gép a tagja és ezek egyetlen megszakításmentes közeghez csatlakoznak nincs is szükség ennél többre. Az adatok hálózati kártyától hálózati kártyáig tudnak benne közlekedni csupán a hálózathozzáférési réteg által nyújtott alapszintű szolgáltatásokra támaszkodva.

Egy útválasztókat is tartalmazó hálózatban ugyanakkor a forgalomirányításhoz már nem elegendő a fizikai címek használata. Azok a felderítési módszerek, amelyek segítségével a gépek a fizikai cím alapján megtalálják egymást egy helyi hálózatban eleve nem is működnek útválasztón keresztül. De még ha működnének, a fizikai címek alapján történő tájékozódás akkor is garantáltan lehetetlen lenne, hiszen a fizikai címek hozzárendelésében semmiféle logikai struktúra nincsen. Bármelyik cím gyakorlatilag bárhol előfordulhat, nincs semmiféle szervező rend a címtérben.

Éppen ezért a TCP/IP elrejti a fizikai címet, a hálózatot pedig olyan logikai címek alapján szervezi meg, amelyek kiosztásában szigorú hierarchia érvényesül. Ezt a logikai címzési sémát az IP protokoll és az internet réteg biztosítja és kezeli. Amint korábban is említettük a logikai alapon kiosztott címeket IP címeknek (IP address) nevezzük. Az internet réteg egy másik fontos protokollja az ARP (*Address Resolution Protocol*). Ez a helyi hálózatról egy olyan táblázatot készít, amely alapján az IP címek gyorsan lefordíthatók fizikai címekké. Ez az úgynevezett ARP tábla tehát a kapocs a fizikai (beégetett) és a logikai (IP) címtér között.

Egy útválasztót is tartalmazó hálózatban (lásd a 4.1. ábrát) a TCP/IP a következő stratégiát követi az adatok továbbítása során:

1. Ha a küldő és a fogadó címük alapján azonos alhálózathoz tartozik, akkor a küldő közvetlenül továbbítja az adatcsomagot a fogadónak. Ilyenkor a fogadó IP címét a rendszer az ARP tábla alapján feloldja fizikai címmé, majd az adatokat a megfelelő hálózati adapternek továbbítja.
2. Ha a címzett egy másik hálózati szegmenshez tartozik, akkor a következők történnek:
   A. A datagramot a küldő először az átjárónak (gateway) küldi el. Az átjáró általánosságban olyan hálózati eszköz, amely képes a forgalmat az egyik szegmensből a másikba továbbítani. (Amint arról korábban, az 1. óra anyagában volt szó, az átjáró alapvetően egy útválasztó szokott lenni.) Ilyenkor a küldő protokollverme az átjáró IP címét oldja fel fizikai címmé az ARP tábla alapján, majd ennek küldi el az adatcsomagot.
   B. A datagram áthalad az átjárón és belép a hierarchiában eggyel magasabb szinten álló hálózati szegmensbe (lásd a 4.1. ábrát), ahol az egész folyamat megismétlődik. Ha a címzett ebben a szegmensben található, az adatok a fizikai cím alapján megérkeznek hozzá. Ha nem, akkor a datagram a következő átjáróhoz kerül.
   C. A datagram végül az átjárók láncolatán keresztül megérkezik abba a hálózati szegmensbe, amelyben a címzett található, itt a címzett IP címe hardvercímmé alakul, az adatokat pedig fogadja a címzett hálózati adaptere.



4.1. ábra

A más hálózathoz tartozó címzetteknek szánt datagramokat először az átjáró fogadja.

Ahhoz, hogy egy összetett, számos útválasztót tartalmazó hálózatban az adatok bárhonnan bárhova továbbíthatók legyenek, az internet réteg protokolljainak a következő funkciókkal kell rendelkezniük:

- A hálózat bármely számítógépét egyértelműen azonosítaniuk kell.
- El kell tudniuk dönteni, mikor kell egy üzenetet az átjárón keresztül továbbítani.
- Rendelkezniük kell egy olyan, hardverfüggetlen eljárással, amely alapján a címzett hálózata azonosítható. Erre azért van szükség, hogy az adatcsomagok a lehető leghatékonyabb úton haladhassanak az útválasztók láncolatán keresztül a megfelelő szegmens felé.
- Rendelkezniük kell olyan módszerrel, amellyel a címzett gép logikai IP címe fizikai címmé fordítható le, amely alapján a célszegmensen belül az adatok a megfelelő hálózati adapterhez továbbíthatók.

Az IP protokoll jelenleg legelterjedtebb változata az IPv4, bár manapság egy az IPv6-ra történő afféle folyamatos átmenetnek lehetünk tanúi. Ebben az órában kizárólag az IPv4 címzési sémájáról lesz szó, valamint bemutatjuk, hogyan továbbítja a TCP/IP az adatcsomagokat egy összetett hálózaton belül az IP és az ARP protokollokra támaszkodva. Szintén szó esik az ugyancsak az internet réteghez tartozó ICMP protokollról, amely a hálózati hibakeresést teszi lehetővé. Az egyelőre alternatívnak számító, de hamarosan várhatóan általánosan elterjedt IPv6 protokollról részletesen majd csak a 13. órában lesz szó.

Az internet réteg az OSI modell hálózati rétegének felel meg, amit gyakran egyszerűen csak a 3. réteg (Layer 3) néven is említenek.

Az Internet Protokoll (IP)

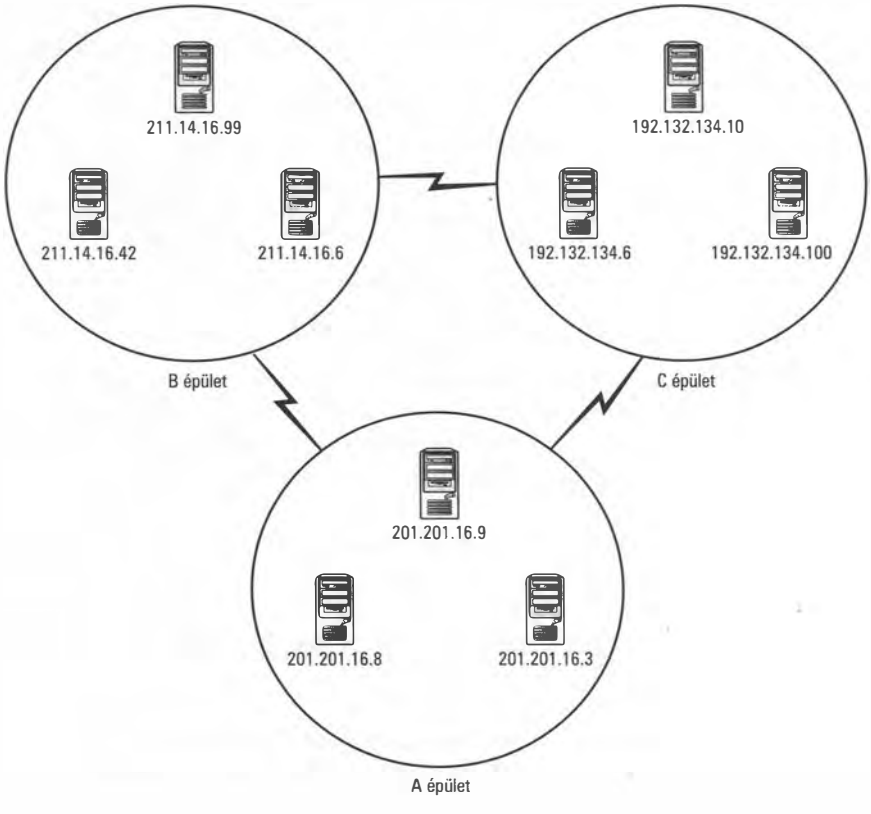
Az IP protokoll egyrészt hierarchikus, hardvertől független címzési rendszert biztosít, másrészt olyan szolgáltatásokat biztosít, melyek révén az adatok eljuthatnak a küldőtől a címzettig egy tetszőlegesen összetett, útválasztókkal tagolt hálózaton keresztül. Egy TCP/IP hálózatban valamennyi hálózati adapternek egyedi IP címmel kell rendelkeznie.

A TCP/IP-vel kapcsolatos leírások gyakran arról beszélnek, hogy maguknak a számítógépeknek van IP címük. Számos esetben valóban ez a helyzet, mivel a legtöbb számítógép csak egy hálózati adapterrel rendelkezik. Ugyanakkor az sem ritka, hogy egy gépben több hálózati csatoló található. Ha egy számítógép útválasztóként vagy proxyként működik, szükségszerűen több hálózati interfésszel kell rendelkeznie, és így garantáltan több IP címe van. Éppen ezért helyesebb azt mondani, hogy az IP cím a hálózati eszközhöz/csatolóhoz tartozik, nem magához a géphez. A hálózati

terminológiában tehát általánosságban a gép vagy gazdagép (host) kifejezés egy olyan hálózati eszközt jelent, amelynek IP címe van, ez az eszköz azonban nem föltétlen a szó szokványos értelmében vett számítógép.

Hogy a helyzet még bonyolultabb legyen egyes operációs rendszerek azt is lehetővé teszik, hogy egyetlen hálózati csatolóhoz több IP cím tartozzon.

Egy hálózatban az IP címek úgy vannak kiosztva, hogy a címekből meg lehessen mondani, nagyjából hol van az a gép, amelyhez a cím tartozik, vagyis melyik hálózatnak vagy alhálózatnak a része (lásd a 4.2. ábrát). Másként fogalmazva az IP cím egy része olyan, mint a közönséges levelek esetében az irányítószám (amely egy általános, tágabb értelemben vett helyet ír le), a másik pedig olyan, mint az utca és a házszám, amely a tágabb területen belüli pontos helyet adja meg.



4.2. ábra

Ha vetünk egy pillantást a címekre, máris kapunk egy átfogó képet a hálózat felépítéséről.

Ha egy ember néz rá a 4.2. ábrára, rögtön látja, hogy az összes olyan cím, amely a 192.132.134 számokkal kezdődik, a C épületben kell legyen. Egy számítógépnek azonban ennél kicsit többre van szüksége ahhoz., hogy boldoguljon. Éppen ezért az IP címeknek két része van:

- Hálózati cím (network ID) és
- Gépazonosító (host ID)

Ahhoz, hogy a gépi tájékozódás működjön a hálózatnak természetesen rendelkeznie kell egy olyan szolgáltatással, amely alapján eldönthető, hogy a címnek melyik része a hálózati és melyik a gépazonosító. Sajnos a hálózatok sokfélesége miatt erre általános megoldás nincsen, vagyis a való élet ezen a ponton kissé bonyolultabb, mint az elmélet. A nagy hálózatoknak nyilván sok bitet kell fönntartaniuk a címekből a gépek azonosítására, míg egy kis hálózatban néhány bit is elegendő arra, hogy egyedi azonosítót kaphasson minden egyes gép. A dolog persze fordítva is igaz, hiszen ha belegondolunk, nyilván sokkal több kis hálózat van, mint nagy, vagyis a kicsik esetében sokkal több bitet igényel a hálózatazonosító.

Amint arról hamarosan szó lesz az eredeti megoldás erre a problémára az volt, hogy az IP címteret címosztályokra (*address class*) bontották. Az A osztályú címekben 8 biten ábrázolták a hálózatazonosítót, a B osztályúakban 16 biten, a C osztályúakban pedig 24-en. Ezt a rendszert aztán később kiegészítették az úgynevezett alhálózatokkal (*subnetting*), amelyek helyi szinten nagyobb szabadságot adtak a hálózat strukturálásában.

Egy ennél is újabb technika a CIDR (*Classless Inter-Domain Routing*), amely tulajdonképpen fölöslegessé tette a hálózati osztályokat. A CIDR címrendszer használata, amely hajlékony, és teljesen egyértelmű módon teszi lehetővé az IP címtér egyes tartományainak kiosztását mára gyakorlatilag általánosnak tekinthető az egész interneten.

Mindazonáltal ha valaki TCP/IP alapú hálózat kiépítésébe szeretne belevágni, egyelőre tisztában kell lennie az osztályalapú és a CIDR típusú címkiosztás rejtelmeivel is. A két technikáról részletesen majd az 5. órában esik szó. Egyelőre elégedjünk meg annyival, hogy a két módszer célja azonos: az IP címeket átlátható módon két részre kell osztaniuk, nevezetesen hálózati címre és gépazonosítóra.

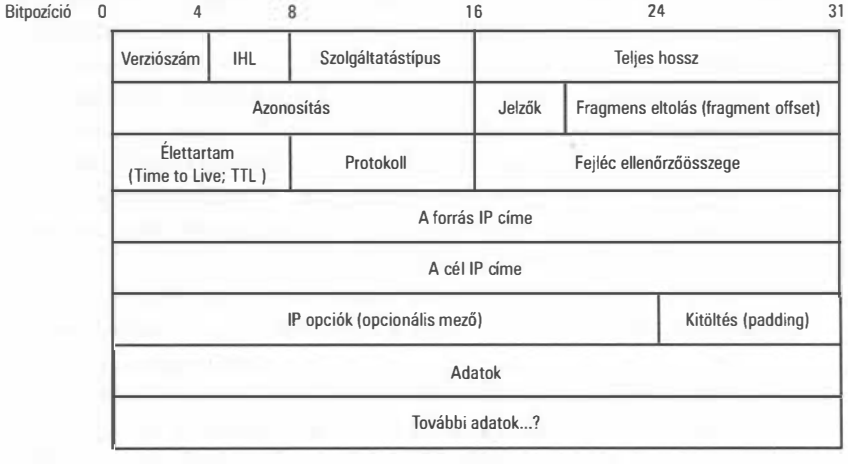
Az 5. óra anyagának olvasása közben majd ne felejtsünk el visszalapozni ehhez a fejezethez is, ha ez szükséges. Amíg az ember meg nem tanulta az alhálózati azonosítók és a CIDR címkiosztás minden fortélyát, addig nem mondhatja el magáról, hogy mestere az IP címzési rendszernek.

Az IP fejléc mezői

Minden IP datagram egy úgynevezett IP fejléccel (*IP header*) kezdődik. Ezt a fejlécet a küldő gépén futó TCP/IP szoftver állítja össze, a fogadó gép pedig az ebben található információk alapján tudja eldönteni, hogy mit kell tennie az adott datagrammal. Az IP fejléc meglehetősen sok információt tartalmaz. Benne van a küldő és a fogadó IP címe, a datagram hossza, a használt IP protokoll verziószáma, valamint néhány speciális, az útválasztóknak szóló utasítás.

Akit bővebben is érdekel, mi minden található egy IP fejlécben, olvassa el az RFC 791-es dokumentumot.

Egy IP fejléc hossza legalább 20 bájt, tartalmát pedig vázlatosan a 4.3. ábra mutatja.



4.3. ábra
Az IP fejléc mezői

Amint a 4.3. ábra is mutatja, az IP fejléc a következő mezőket tartalmazza:

- **Verziószám** – Ez a 4 bites mező mutatja, hogy az IP protokoll melyik verzióját használjuk az adott kommunikációban. A jelenleg használatos verziószám a 4-e, aminek a bináris 0100 bitminta felel meg.
- **IHL (*Internet Header Length*)** – Ez az ismét csak 4 bites mező az IP fejléc 32 bites szavakban mért hosszát adja meg. A fejléc legalább öt ilyen 32 bites szóból kell álljon, vagyis ennem a mezőnek a legkisebb értéke a bináris 0101.
- **Szolgáltatástípus (*Type of Service*)** – Az útválasztók által igényelt információt tulajdonképpen maga az IP cím is tartalmazza, így egyes útválasztók egyszerűen figyelmen kívül hagyják ennek a mezőnek a tartalmát. Ugyanakkor a mostanában egyre népszerűbb

QoS (Quality of Service) szolgáltatások erre a mezőre alapozzák a működésüket, vagyis az itt található információ mégsem fölösleges. Ennek a 8 bites mezőnek tulajdonképpen az a célja, hogy segítségével valamiféle prioritási sorrendet lehessen fölállítani azon datagramok között, amelyek egy útválasztónál átbocsátásra várakoznak. Ezzel együtt az IP protokoll legtöbb mai megvalósítása csupa nullát helyez el ebben a mezőben.

- Teljes hossz (*Total length*) – Ez a 16 bites mező tartalmazza a datagram 8 bites egységekben (octet) mért teljes hosszát, vagyis a fejléc és a szállítandó hasznos adatok hosszának összegét.
- Azonosítás (*Identification*) – Ez a 16 bites mező egy automatikusan inkrementálódó sorozatszámot tartalmaz, amely a forrás IP által küldött üzeneteket azonosítja. Ha az IP réteg egy olyan üzenetet kap, amit nem tud elhelyezni egyetlen datagramban, akkor azt megfelelően kis részekre tördeli és minden rész fejlécében ugyanazt az azonosítót helyezi el. A fogadó e sorozatszámok alapján fogja tudni, hogy mely fragmenseket kell összeillesztenie egész datagrammá.
- Jelzők (*flags*) – Ez a mező tartalmazza azt az információt, hogy a fragmentálással kapcsolatos lehetőségek közül melyeket használt a rendszer az átvitel során. A mező első bitjét nem használjuk, ennek értéke definíció szerint nulla. A következő bit az úgynevezett DF (*Do not Fragment*) bit. Ha a DF bit értéke 0, akkor a fragmentálás engedélyezett, egyébként nem. A következő az MF (*More Fragments*) bit, ami azt jelzi a fogadónak, hogy további fragmensek fognak még érkezni, vagyis várjon az összeillesztéssel. Ha az MF bit értéke 0, az a fogadó számára azt jelenti, hogy vagy egyáltalán nem történt fragmentálás az átvitel során, vagy már minden fragmens megérkezett.
- Fragmens eltolás (*Fragment Offset*) – Ez egy 13 bit hosszúságú mező, amely az egymás követő fragmensek sorszámát tartalmazza. A fogadó ez alapján képes újra összerakni a részekből a teljes üzenetet. A fragmens eltolás mezőben található érték a szabvány szerint 8 bites egységekben mért eltolást jelent.
- Élettartam (*Time to Live; TTL*) – Ennek a mezőnek az értéke határozza meg, hogy a hálózati eszközöknek hány másodpercig, vagy hány ugrásig (*router hop*) kell próbálkozniuk a datagram továbbításával. Amelyik datagramnak lejár az élettartama, azt a rendszer egyszerűen eldobja. Minden útválasztó megvizsgálja ennek a mezőnek az értékét, és a datagramot úgy továbbítja, hogy az értéket vagy eggyel csökkenti, vagy annyival, ahány másodpercet a csomag az útválasztóban várakozott. Ha az érték eléri a nullát, a csomag megsemmisül.
- Az ugrásszám (*hop; router hop*) azt mutatja meg, hogy az átvitel során a csomagnak hány útválasztón kellett áthaladnia. Ha a datagramnak mondjuk öt útválasztót kell érintenie, mielőtt megérkezne a rendeltetési helyére, akkor azt mondjuk, hogy öt ugrást tartalmazott az átvitel.
- Protokoll – Ez a 8 bites mező azonosítja azt a protokollt, amely a fogadó oldalon megkapja a kérdéses datagramot. Ha például a mező értéke decimális 6 (vagyis bináris 00000110), akkor a fogadó gép TCP modulja fogja megkapni. A következő táblázat néhány általánosan használt protokoll azonosítóját tartalmazza:

Protokoll neve	Protokoll azonosítója
ICMP	1
TCP	6
UDP	17

- Fejléc ellenőrzőösszege (*Header checksum*) – Ez a mező egy 16 bites számított értéket tartalmaz, amely alapján eldönthető, hogy a fejléc megsérült-e az átvitel során. (Fontos hangsúlyozni, hogy itt csupán a fejléc épségéről van szó.) Az ellenőrzőösszeget minden egyes útválasztó újraszámolja, mikor csökkenti a TTL értékét, hiszen ez változást okoz a fejlécben.
- A forrás IP címe (*Source IP Address*) – 32 bites mező, amely a datagram forrásának IP címét tartalmazza.
- A cél IP címe (*Destination IP Address*) – Szintén 32 bites mező, amely a célgép IP címét tartalmazza. A fogadó e mező alapján tudja ellenőrizni, hogy az üzenet valóban neki szólt.
- IP opciók (*IP options*) – Ez a mező néhány olyan beállítást tesz lehetővé, amelyeknek a tesztelés és a hibakeresés során van jelentősége, illetve amelyek a hálózati biztonsággal kapcsolatosak. Ilyen például a kötött útvonal (*Strict Source Route*) amely arra utasítja a datagramot, hogy az útválasztók egy adott láncolatát járja be, vagy az internetes időbélyeg (*Internet Timestamp*) amelynél minden egyes útválasztó, amelyen áthalad a datagram, időbélyeget fűz hozzá. Egyes biztonsági korlátozások is ezen a mezőn keresztül léptethetők életbe.
- Kitöltés (*Padding*) – Az IP opciókat tartalmazó mező hossza természetéből adódóan változhat. Ugyanakkor a fejléc teljes hossza 32 egész számú többszöröse kell maradjon, mert az IHL mező értéke 32 bites szavakban mérve tartalmazza azt. Ez úgy oldható meg, hogy ebben a mezőben a rendszer elhelyezi az ehhez szükséges számú 0 bitet.
- Adatok (*IP data Payload*) – Ez a mező általában a szállítási réteghez tartozó TCP, vagy UDP protokoll számára továbbítandó adatokat tartalmaz, illetve szólhat a tartalma az ICMP vagy az IGMP protokoll számára is. Az adatok mennyisége változó, de általában néhány ezer bájt.

Az IP címzés rendszere

Az IP cím egy 32 bites bináris szám. Ez a 32 bites cím logikailag négy, egyenként nyolc bites részre, úgynevezett oktetre (*octet*) tagolódik. Mivel azonban mi emberek nem igazán tudunk 32 bites bináris számokkal dolgozni, sőt a legtöbbünknek még a nyolc bites darabok is megfekszik a gyomrát, az IP címeket az esetek túlnyomó többségében négy, egymástól pontokkal elválasztott decimális szám formájában (*dotted decimal format*) szokás megadni. Ebben a formában minden egyes nyolcbites darabot egyenként fordítanak le decimális megfelelőjükké. Ez az a bizonyos négy tízes számrendszerbeli szám (4 x 8 = 32 bit), amelyeket aztán pontokkal elválasztva egymás mellé írnak.

Nyolc biten 0 és 255 között bármely egész szám ábrázolható, vagyis egy IP cím elvileg négy, ebbe a tartományba eső egész számból állhat. A korábbi fejezetekben amúgy már használtunk ilyen címeket, illetve találkozhatott velük az olvasó bármely a TCP/IP protokollal foglalkozó dokumentációban is. Egy decimális formában megadott érvényes IP cím tehát valahogy így fest: 209.121.131.14.

Az IP cím egy részét, a hálózat (*network ID*), más részét a konkrét számítógép azonosítására (*host ID*) használjuk. Amint arról korábban már volt szó, a hálózati és gépazonosító elkülönítésére az eredeti séma a hálózati osztályok kijelölése volt. Bár a mostanában egyre több helyen használt osztálymentes CIDR címzés csökkentette az osztályok jelentőségét, azért az eredeti címzési séma még kellően sok helyen tartja magát ahhoz, hogy egy ilyen könyvből, mint ez, ne lehessen egyszerűen kihagyni az ismertetését. Ráadásul ennek a régi módszernek az elve a TCP/IP címzési rendszerének afféle sarkköve is, tehát kiváló kiindulási pontul szolgálhat számunkra. Az IP címzési technikákról amúgy az 5. órában ennél sokkal részletesebben is lesz szó, most azonban nézzük a címosztályok rendszerét.

Az IP címosztályok rendszerében a teljes IP címtér részekre van osztva. A felosztás szerint a legtöbb cím a következő osztályok valamelyikébe esik:

- **A osztályú (Class A) címek** – A hálózati címet az IP cím első 8 bitje jelenti. A maradék 24 bit a számítógépet azonosítja.
- **B osztályú (Class B) címek** – A hálózatot az első 16 bit azonosítja. Az IP cím másik 16 bitje a számítógépet írja le.
- **C osztályú (Class C) címek** – A hálózat azonosítására az első 24 bit szolgál, míg a gépet a maradék 8 bit azonosítja.

A több bit értelemszerűen többféle bitkombinációt jelent, vagyis – ahogy azt bizonyára az olvasó is kitalálta – az A címosztályba viszonylag kevés hálózat tartozhat, ellenben egy-egy ilyen hálózatban rengeteg gép lehet. Kicsit konkrétabban egy A címosztályba tartozó hálózatban körülbelül 2^{24}, vagyis 16777216 különböző cím osztható ki a gépeknek. Egy C osztályú hálózathoz ezzel szemben egészen kevés, mindössze 2^8 számítógép tartozhat, ugyanakkor a sok bites hálózatazonosítónak köszönhetően rengeteg ilyen kis hálózat létezhet. (A gépek legnagyobb száma valójában csak 254, mivel az elvileg használható 256-ból le kell vonni a csupa nullákból és a csupa egyesekből álló két címet.)

Mármost az olvasó nyilván azon töpreng, vajon mégis honnan tudja egy útválasztó, hogy egy adott címet A, B vagy éppen C osztályúként értelmezzen-e. A megfejtés egyszerű: a TCP/IP tervezői úgy állapították meg a címek használatának szabályait, hogy az osztály magából a címből következzen. A bináris cím első néhány bitje egyértelműen meghatározza, hogy az adott cím melyik osztályhoz tartozik (lásd a 4.1. Táblázatot). A címek osztályba sorolásának szabály a következők:

- Ha a 32 bites cím első bitje 0, akkor garantáltan A osztályú címről van szó.
- Ha a bináris cím az 10 bitkombinációval kezdődik, akkor B osztályú.
- Ha az IP cím első három bitje az 110 kombináció, akkor az illető cím a C osztály tagja.

Ezt a sémát (szerencsére) decimális formára is egész egyszerűen át lehet ültetni, hiszen a fenti szabályok csak az első számot befolyásolják. Például egy A osztályú cím első bitje mindenképpen nulla kell legyen, vagyis egy ilyen cím decimális alakjában az első tag garantáltan nem lehet nagyobb mint 127. A bináris és decimális címek egymásba alakításáról ebben az órában még részletesebben is lesz szó. Egyelőre elegendő, ha vetünk egy pillantást a 4.2. Táblázatra, amelyben összefoglaltam az A, B és C osztályú címtartományokat. Figyeljük meg, hogy egyes címtartományok nem írnak le valós hálózatokat. Ezeket speciális célokra tartják fönn. A speciális IP címekről szintén ejtünk még szót ebben az órában.

4.1. Táblázat *Az A, B és C osztályú IP címek kiosztása*

Címosztály	A bináris cím kezdete	A decimális cím első tagja	Fönntartott címek
A	0	0-tól 127-ig	10.0.0.0-tól 10.255.255.255-ig; 127.0.0.0-tól 127.255.255.255-ig
B	10	128-tól 191-ig	172.16.0.0-tól 172.31.255.255-ig
C	110	192-től 223-ig	192.168.0.0-tól 192.168.255.255-ig

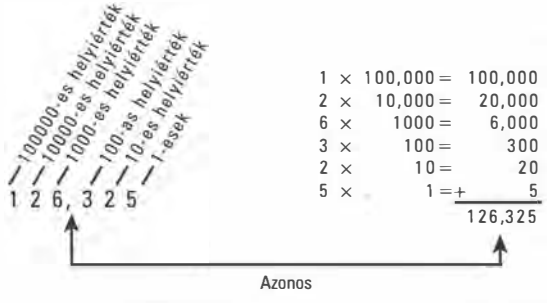
Az internet specifikációjában a fentieken kívül leírnak D és E osztályú címeket is. A D osztályú címeket részleges üzenetszórásra (*multicast*) használják. Ilyenkor az üzenetet a küldő a hálózat egy kijelölt részének, de nem az egész hálózatnak címzi. Létezik természetesen egyszerű üzenetszórás (*broadcast*) is, amikor a címzett a hálózat valamennyi gépe. Egy D osztályú (Class D) cím esetében az első négy bitje az 1110 mintát tartalmazza. Decimális címekben gondolkodva ez azt jelenti, hogy a cím első tagja a 224-től 239-is terjedő számtartományba esik. Az E osztályú (Class E) címeket kísérleti jellegű feladatokra használják, így normál munkakörnyezetben soha nem bukkannak föl. Egy ilyen cím bináris alakjában az első öt bit az 11110 bitmintát tartalmazza, ami a decimális 240-247 tartományt jelenti az első tagban.

Egy hálózat tulajdonosa dönthet úgy, hogy a rendelkezésére álló címtartományt kisebb részekre, úgynevezett alhálózatokra (*subnets*) bontja. Ez praktikusan azt jelenti, hogy a gépet azonosító bitek közül föláldozunk néhány továbbit az alhálózatok oltárán, és hozzácsapjuk azokat a hálózatazonosító részhez. Amint azt bizonyára az olvasó is kitalálta ez a módszer elsősorban az A és B osztályú címtartományoknál használatos, hiszen ott jócskán van miből biteket elvenni. Ez persze nem jelenti azt, hogy C osztályú hálózatban ne lehetne alhálózatokat kijelölni, csak ott jóval kisebb a mozgástér. Az alhálózatok kezeléséről bővebben az 5. órában lesz szó.

Elméletileg minden az internetre kapcsolódó számítógépnek egyedi IP címmel kellene rendelkeznie. A gyakorlatban ugyanakkor a proxy szerverek és a NAT (*Network Address Translation*) használata révén nem egyedi címekkel rendelkező gépek is tudnak az internet át kommunikálni. A NAT-képes eszközökről bővebben a 12. órában ejtünk majd szót.

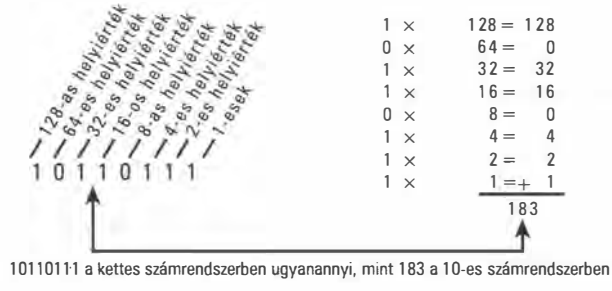
32 bites bináris címek decimális címmé alakítása

A kettes számrendszerbeli számok tulajdonképpen ugyanolyanok, mint a köznapi életben használt tízes számrendszerhez tartozók. Az egyetlen lényeges eltérés az, hogy a helyiértékek itt nem 10, hanem 2 hatványai. Amint az a 4.4. ábrán is látható egy tízes számrendszerbeli egész szám jobbról számított első jegye az egyeseket tartalmazza, aztán jobbra haladva minden egyes helyiérték az előző tízszerese. A decimális szám tényleges értékét úgy kapjuk, hogy összeadjuk az egyes helyiértékek így kiszámított értékeit. Példának okáért a decimális 126325 értéke a következőképpen számítható ki: (1 x 100000) + (2 x 10000) + (6 x 1000) + (3 x 100) + (2 x 10) + (5 x 1) = 126325.



4.4. ábra

A tízes számrendszer használata



4.5. ábra

A bináris (kettes) számrendszer használata

A kettes számrendszerbeli számok esetében tulajdonképpen teljesen azonos a szisztéma. Az egyesek itt is a szám jobb oldalán találhatók, balra haladva pedig minden érték kettővel többet jelent, mint az előző (lásd a 4.5. ábrát).

A számítógépek azért használják kivétel nélkül a kettes számrendszert, mert ebben csak egyesek és nullák fordulhatnak elő, ezek pedig egyszerűen megfeleltethetők az áramkörök kikapcsolt illetve bekapcsolt állapotának.

Ha egy bináris szám decimális megfelelőjét (értékét) szeretnénk meghatározni, semmi egyebet nem kell tennünk, csak összeadni mindazon helyiértékeket, amelyeknél 1-es van a binárisan ábrázolt számban. Emlékezzünk vissza, hogy az IP címek 32 bitből állnak, amit logikailag négy 8 bites számra (oktetre) bontunk, és ezeket egyenként számoljuk át tízes számrendszerbeli megfelelőjükké. Így keletkezik az IP cím pontokkal elválasztott decimális formája (*dotted decimal format*). Lássunk erre rögtön egy példát. Legyen a konvertálandó 32 bites bináris IP cím a következő:
01011001000111011100110000011000.

Ha ezt decimális címmé akarjuk alakítani, a következőket kell tennünk:

1. Először is bontsuk föl a 32 bitet 8 bites egységekre:
   oktet : 01011001
   oktet : 00011101
   oktet : 11001100
   oktet : 00011000
2. Alakítsunk át minden 8 bites bináris számot a decimális megfelelőjévé. Ezt a folyamatot a 4.2. Táblázat szemlélteti.

4.2. Táblázat *Bináris cím decimálissá alakítása*

Oktet	Bináris érték	Átszámítás	Decimális érték
1	01011001	1+8+16+64	89
2	00011101	1+4+8+16	29
3	11001100	4+8+64+128	204
4	00011000	8+16	24

3. Balról jobbra haladva írjuk egymás mellé a keletkezett decimális számokat, közéjük pedig tegyünk pontokat. A kérdéses IP cím tehát a következő: 89.29.204.24.

Ha valaki még gyakorolni szeretné a bináris címek decimálisra történő átszámítását, lapozzon a gyakorlatok részhez.

Decimális számok bináris oktetté alakítása

A decimális számok bináris alakra hozása során tulajdonképpen a 4.5. ábrán bemutatott szisztéma fordítottját kell megvalósítanunk. Ha tehát egy pontozott decimális formában megadott IP címet szeretnénk bináris alakra hozni, akkor először az egyes, pontokkal elválasztott értékeket kell kettes számrendszerbe átszámítanunk, majd az így keletkezett bitnyolcasokat ugyanebben a sorrendben egymás után kell fűznünk. A következőkben példaként bemutatjuk, hogyan lehet a decimális 207-et átszámítani bináris oktetté.

A példában feltételezzük, hogy a decimális szám, amit át szeretnénk számítani kettes számrendszerbe egy IP címből származik, vagyis kisebb, mint 255. Ha ennél nagyobb szám bináris megfelelőjét keressük, a folyamatot természetesen akkor is így kell végrehajtani, de a 4.5. ábrán látható helyiértékek sorozatát szükség szerint ki kell egészítenünk nagyobbakkal is.

A decimális 207 bináris oktetté alakítását a következő műveletek segítségével valósíthatjuk meg:

1. Hasonlítsuk össze az átalakítani kívánt decimális számot (jelen esetben a 207-et) 128-cal. Ha az átalakítandó szám a nagyobb, vagy esetleg éppen 128, akkor vonjunk le belőle 128-at, és írjunk le egy 1-est. Ha az átalakítani kívánt szám kisebb, mint 128, akkor ne vonjunk ki belőle semmit és írjunk le egy 0-át.
   207 > 128
   207 – 128 = 79
   A 128-as helyiértékre beírunk egy 1-est.
   Egyelőre ott tartunk hogy: 1
2. Vegyük az első lépésben keletkezett maradékot (jelen esetben 79) és hasonlítsuk össze 64-gyel. Ha nagyobb, mint 64, akkor vonjuk le belőle ezt az értéket, és írjunk le egy 1-est. Ha kisebb, mint 64, ne vonjunk ki belőle semmit, és írjunk le egy 0-át.
   79 > 64
   79 – 64 = 15
   Leírunk egy 1-est a 64-es helyiértékre.
   Az eredmény jelenleg: 11
3. Vegyük az előző lépés maradékát (jelen esetben 15) és hasonlítsuk össze a 32-vel. Ha nagyobb nála, akkor vonjunk le belőle 32-t és írjunk le egy egyest. Ha a maradék kisebb, mint 32, akkor nem kell levonni belőle semmit, a bináris érték pedig egy 0-val hosszabbodik.
   15 < 32
   15 – 0 = 15
   A 32-es helyiértékre beírunk egy 0-át.
   Az eredmény jelenleg: 110

4. A 3. lépés maradékát (még mindig 15) hasonlítsuk össze 16-tal. Ha nagyobb nála, akkor vonjunk le belőle 16-ot és írjunk le egy 1-est. Ha kisebb, akkor az eljárás ugyanaz, mint korábban, vagyis nullát vonunk le, és nullát írunk az eredménybe.
   15 < 16
   15 – 0 = 15
   A 16-os helyiértéken 0 lesz.
   Az eredmény jelenleg: 1100
5. A 4. lépés maradékát (továbbra is 15) hasonlítsuk össze 8-cal. Ha nagyobb, mint 8, akkor vonjuk le belőle ezt a heliyértéket, az eredmény végére pedig írjunk egy 1-est. Ha kisebb, akkor nullát kell kivonni, és az eredmény végére is 0 kerül.
   15 > 8
   15 – 8 = 7
   A 8-as helyiértékre 1 kerül.
   Az eredmény jelenleg: 11001
6. Az 5. lépés maradékát (7) hasonlítsuk össze 4-gyel. Ha nagyobb nála, akkor vonjunk le belőle 4-et és az eredmény végéhez írjunk hozzá egy 1-est. Ha nem, akkor 0-át kell levonni, és 0 lesz az adott helyiértéken.
   7 > 4
   7 – 4 = 3
   A 4-es helyiértéken 1 lesz.
   Az eredmény jelenleg: 110011
7. A 6. lépésben keletkezett maradékot (3) hasonlítsuk össze 2-vel. Ha nagyobb nála, akkor vonjunk le belőle kettőt, az eredményhez pedig írjunk hozzá egy egyest. Ha nem, akkor nem kell az értéket csökkenteni, az eredmény következő jegye pedig 0.
   3 > 2
   3 – 2 = 1
   A kettes helyiértéken 1 áll.
   Az eredmény jelenleg: 1100111
8. Ha a 7. lépésben keletkezett maradék 1, akkor az eredményhez írjunk hozzá egy 1-est. Ha a korábbi maradék 0, akkor az eredmény utolsó jegye is nulla.
   1 = 1
   Hozzáírunk egy egyest a bináris számhoz.
   A végeredmény: 11001111

Átalakítottuk tehát a decimális 207-et annak bináris alakjává, és az eredmény 11001111 lett.

Speciális IP címek

Egyes IP címek speciális jelentéssel bírnak, így egyetlen konkrét számítógép sem kaphatja meg ezeket. A csupa nullából álló gépazonosító a hálózat egészére vonatkozik. A 129.152.0.0 cím például egy teljes B osztályú hálózatot címez meg, amelynek hálózatazonosítója 129.152.

A csupa egyesből álló gépazonosító úgynevezett üzenetszórást jelöl. Az ilyen üzenet (*broadcast*) a hálózat valamennyi gépéhez eljut, és valamennyinek fel is kell dolgoznia azt. A 129.152.255.255 IP cím tehát egy üzenetszórási cím (*broadcast address*), amely a teljes 129.152 hálózatazonosítójú (B osztályú) hálózatot jelenti, annak összes tagjával. (Egy megjegyzés: a csupa egyesekből álló oktetnek a decimális alakban a 255 felel meg.)

A 255.255.255.255 cím szintén használható üzenetszórásra.

A 127-tel kezdődő címek az úgynevezett visszacsatolási címek (*loopback address*). A visszacsatolási címre küldött üzenetet tulajdonképpen a helyi TCP/IP szoftver küldi önmagának. Ez a TCP/IP rendszer működésének ellenőrzésére szolgáló egyik módszer. Erről és a `ping` parancsról a 14. órában még részletesen esik majd szó. A leggyakrabban használt visszacsatolási cím a 127.0.0.1.

Az RFC 1597-ben található specifikáció bizonyos IP címtartományokat magáncélú felhasználásra tart fönn. Ezzel kapcsolatban az az alapfeltevés, hogy az ilyen címmel rendelkező gépek nem csatlakoznak közvetlenül az internethez, vagyis nem kell globálisan egyedi címmel rendelkezniük. Manapság gyakran találkozhatunk olyan védett hálózatokkal, amelyekben ilyen privát címeket használnak, az internetet pedig egy NAT-ra (*Network Address Translation*) képes eszközön keresztül érik el. Erről a lehetőségről majd a 12. órában lesz szó bővebben. A privát hálózati címtartományok a következők:

- 10.0.0.0-tól 10.255.255.255-ig
- 172.16.0.0-tól 172.31.255.255-ig
- 192.168.0.0-tól 192.168.255.255-ig

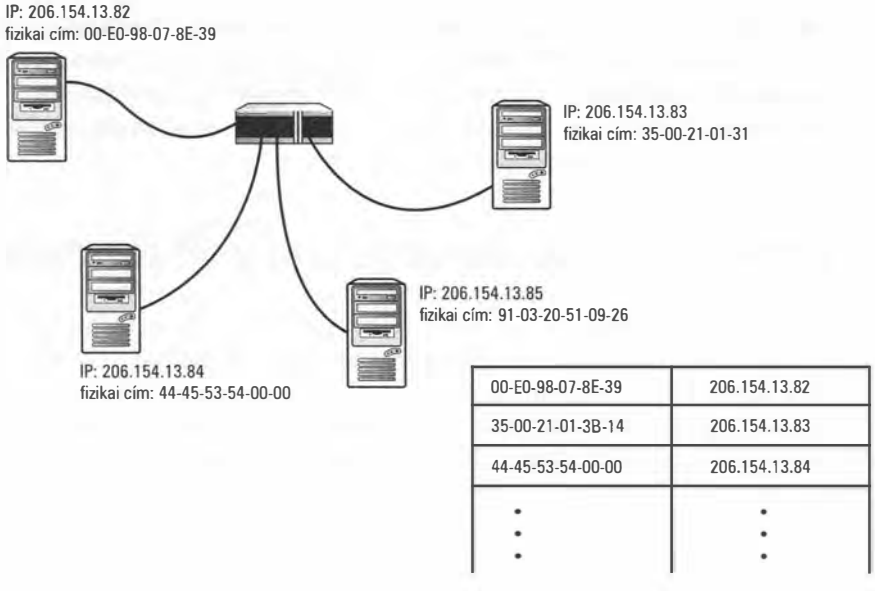
Mivel a privát címeket nem kell egyeztetni a világ többi hálózatával, a fenti tartományok mindegyike teljes egészében rendelkezésére áll az összes ilyen magánhálózatnak. A címek kiosztása, vagy az alhálózatok kijelölése az ilyen helyeken egyedül a hálózati adminisztrátor elhatározásától függ, akinek így értelemszerűen szinte korlátlan lehetőségei vannak a hálózat strukturálása terén.

A 169.254.0.0tól 169.255.255.255-ig terjedő címeket az úgynevezett autokonfigurációs szolgáltatások számára vannak fönntartva. A Zeroconf rendszerről és az automatikus hálózati beállításokról bővebben a 12. órában esik majd szó.

Az ARP (Address Resolution Protocol) protokoll

Amint arról korábban már volt szó, egy helyi hálózat számítógépei egy az internet réteghez tartozó protokoll, az **ARP** (*Address Resolution Protocol*) segítségével rendelik egymáshoz a fizikai címeket és a logikai IP címeket. Ahhoz, hogy egy helyi hálózat egyik gépe üzenetet tudjon küldeni egy másiknak, ismernie kell a címzett hálózati adapterének fizikai címét. Éppen ez adja az ARP protokoll fontosságát. Ugyanakkor a TCP/IP megvalósításai általában teljesen elrejtik a felhasználó elől a kommunikáció fizikai részleteit, vagyis bármennyire is fontos, a közönséges felhasználó gyakorlatilag semmit nem lát az ARP működéséből. Számára hálózati csatolót az IP cím azonosítja. Mindez persze semmit nem változtat azon a tényen, hogy a színfalak mögött ezt a címet mindenképpen hozzá kell rendelni egy fizikai címhez, máskülönben az üzenet nem fog eljutni a címzetthez.

A hálózati szegmens valamennyi számítógépe fönntart a memóriájában egy **ARP táblának**, vagy **ARP gyorstárnak** (*ARP cache*) nevezett táblázatot. Az ARP gyorstár az adott alhálózatban található gépek fizikai és logikai (IP) címeinek megfeleltetését tartalmazza (lásd a 4.6. ábrát). Ha egy számítógép üzenetet akar küldeni egy ugyanazon a szegmensen található másik gépnek, akkor annak fizikai címét először az ARP gyorstárban keresi. Az ARP gyorstár frissítése dinamikusan történik. Ha a gép nem találja a szükséges címet a táblázatban, akkor üzenetszórással kiküld egy **ARP kérést** (*ARP request frame*) a teljes alhálózatba.



4.6. ábra

Az IP címek és a fizikai címek összerendelését az ARP végzi.

Ez a mindenkinek kiküldött ARP kérés a helyi gép által feloldani kívánt IP címet tartalmazza, illetve benne van annak a gépnek a fizikai címe is, akitől a kérdés származik. A szegmens gépei megkapják az üzenetet, kiolvassák belőle a feloldandó IP címet, összehasonlítják a sajátjukkal, és az, amelyik a saját IP címét találja a kérésben, visszaküldi a fizikai címét a kérdezőnek. Ez az újonnan feloldott IP cím fizikai cím páros természetesen bekerül az ARP táblába, vagyis a következő alkalommal ezt a folyamatot már nem kell végigjátszani.

Az ARP tábla bejegyzései egy bizonyos előre meghatározott idő után elévülnek. Az elévült bejegyzéseket a rendszer törli a gyorstárból, így amikor ismét szükség lenne rájuk, megint lefut a címfeloldási folyamat , és a táblába friss bejegyzés kerül.

A fordított ARP (RARP)

A RARP (Reverse ARP) protokoll épp a fordítottját teszi annak, mint ami az ARP protokoll rendeltetése. Az ARP protokollt akkor használja a rendszer, ha egy ismert IP címhez keresi a megfelelő fizikai címet. A RARP ezzel szemben ismert fizikai cím alapján képes IP címet szolgáltatni. A RARP leggyakoribb felhasználási területe a merevlemez nélküli munkaállomások üzemeltetése. Ilyenkor a BOOTP protokollal együtt használják.

BOOTP (Boot PROM)
Számos hálózati csatolón találunk egy üres foglalatot, ahova egy boot PROM-nak nevezett integrált áramköri elemet lehet csatlakoztatni. A boot PROM-ban található program (firmware) a számítógép bekapcsolásakor azonnal elindul. Rendeltetése az, hogy egy hálózati kiszolgálóról töltsön be egy operációs rendszert és ezzel indítsa el a helyi gépet. A gép tehát ilyenkor nem a helyi merevlemezről, hanem hálózatról bootol, a BOOTP eszközre letöltődő operációs rendszer pedig egy előre megadott IP címet rendel a hálózati csatolóhoz.

Az ICMP (Internet Control Message Protocol) protokoll

Egy távoli számítógépnek küldött adatok nem ritkán útválasztók egész láncolatán haladnak keresztül, amíg megérkeznek rendeltetési helyükre. Az útválasztók működésével kapcsolatban menet közben számos különféle gond adódhat. Ezekről a hibákról az útválasztó az ICMP (*Internet Control Message Protocol*) segítségével értesíti a küldőt. Az ICMP-t emellett diagnosztikai célokra és hibakeresésre is használják.

A leggyakoribb ICMP üzenettípusokat a következő felsorolás tartalmazza. Talán érdemes megjegyezni, hogy az itt felsoroltakon kívül még jó néhány egyéb olyan állapot is létezik, amely valamilyen ICMP üzenet kiváltását okozza, de ezek az események egészen ritkák.

- **Visszhang kérés és visszhang válasz (*Echo Request; Echo Reply*)** – Az ICMP-t gyakran használják tesztelésre. Az a hálózati adminisztrátor, aki a `ping` parancs segítségével ellenőrzi, hogy egy távoli gép elérhető-e a hálózat adott pontjáról az ICMP protokollt használja. A `ping` ugyanis egy olyan csomagot küld ki a vizsgált IP címre, amit az ott található gépnek vissza kell küldenie. A `ping` program ehhez az ICMP protokoll `Echo Request` és `Echo Reply` parancsait használja.
- **Forráslassítás (*Source Quench*)** – Ha egy gyors számítógép nagy mennyiségű adatot küld egy másiknak, az adatmennyiség akkora lehet, hogy azt valamelyik útválasztó nem tudja kellő sebességgel feldolgozni. Ilyenkor visszaküld a forrásnak egy `Source Quench` ICMP üzenetet, amivel arra utasítja, hogy lassítson le arra a sebességre, ami számára is elfogadható. Amennyiben az szükséges, további ilyen üzeneteket is küldhet az útválasztó az adatforrásnak.
- **A célgép nem elérhető (*Destination Unreacheable*)** – Ha egy útválasztó olyan datagramot kap, amit nem tud kézbesíteni, akkor az ICMP protokoll egy `Destination Unreacheable` üzenetet küld vissza a feladónak. Ilyesmi például akkor történhet, ha e célhálózat átjáróját karbantartás miatt lekapcsolták.
- **Időtúllépés (*Time Exceeded*)** – Az ICMP ezt az üzenetet akkor küldi el a forrásnak, ha egy datagram TTL értéke út közben eléri a nullát. Ez több dolgot jelenthet. Elképzelhető, hogy a datagramnak egyszerűen csak túl sok útválasztón kell keresztülhaladnia ahhoz, hogy a célt elérje, ez pedig a jelenlegi TTL beállításokkal nem lehetséges. Ugyanakkor a jelenség utalhat az útválasztási táblák olyan hibájára is, amelynek következtében a csomag többször ugyanazt a hurkot járja be.

Ilyen útválasztási hurok keletkezik, ha az átvitelek sorozata visszavezet egy olyan útválasztóhoz, amit a datagram korábban már érintet. Képzeljük el, hogy van egy-egy útválasztó Los Angelesben, San Franciscoban és Denverben. A Los Angelesben működő gép elküldi a datagramot San Fancoscoba, az továbbküldi Denverbe, A denveri viszont egy beállítási hiba miatt visszaküldi Los Angelesbe. A datagran ilyenkor csapdába esik, és vég nélkül köröz e három résztvevő között, amíg a TTL értéke nullára nem csökken. Útválasztási huroknak az interneten elvileg nem szabadna keletkeznie, de azért néha szokott. Ennek a leggyakoribb oka az, amikor egy hálózati adminisztrátor statikus bejegyzéseket fűz hozzá az útválasztási táblához, de elnéz valamit.

- **Fragmentálás kérése (*Framentation Needed*)** – Ez az ICMP üzenet akkor megy vissza a küldőnek, ha egy útválasztó olyan datagramot kap, amelyben be van állítva a fragmentálás tiltását jelző bit (Do Not Fragment bit), neki azonban fragmentálnia kellene, mert csak így tudja továbbküldeni a következő útválasztónak.

Az internet réteg egyéb protokolljai

Az internet réteghez a felsoroltakon kívül tartozik még néhány más protokoll is. Ezek közül a BGP (*Border Gateway Protocol*) és a RIP (*Routing Information Protocol*) az útválasztók működéséhez szükséges. Róluk a 8. órában az útválasztás részletes tárgyalása során lesz majd szó bővebbem.

Az IPsec protokollok az IPv4-ben még csupán opcionális elemek, az IPv6-ban azonban már kötelező lesz a használatuk. Ezekről a szolgáltatásokról, melyek biztonságos, titkosított kommunikációt tesznek lehetővé majd a 23. órában esik szó bővebben. Egyes az internet réteghez tartozó protokollok a részleges üzenetszórással (*multicasting*) kapcsolatosak. Végezetül ismét felhívjuk a figyelmet arra a korábban már említett tényre, mely szerint az internet réteghez tartozó protokollokat az OSI modell után gyakran nevezik röviden Layer 3 protokolloknak is a szakirodalomban.

Összefoglalás

Ebben az órában az internet réteg protokolljairól (IP, ARP, RARP, ICMP) volt szó. Az IP olyan a hardvertől független logikai címzési sémát biztosít, amely lehetővé teszi, hogy az adatok a hálózat bármely pontjáról bármely más helyre eljuthassanak. Megismerkedtünk az IP címeknek a bináris és decimális formájával, illetve azok osztályba sorolásával (A, B, C, D és E osztályú címekről volt szó). Megtanultuk az ARP és a RARP működésének alapjait, így már tudjuk, hogy az első az ismert IP címet képes fizikai címhez hozzárendelni, míg a második épp ennek az ellenkezőjét teszi, így elsősorban merevlemez nélküli munkaállomások üzemeltetése során használják. Végezetül az ICMP protokollt elsősorban tesztelési és hibakeresési célokra használják.

Kérdések és válaszok

K *Milyen általánosan elfogadott jelölésrendszerrel egyszerűsíthető a 32 bites bináris címek használata?*

V Ez az úgynevezett pontozott decimális forma (*dotted decimal format*).

K *Milyen információt ad vissza az ARP protokoll, ha egy ismert IP címet adunk át neki?*

V A kérdéses IP címmel rendelkező hálózati csatoló fizikai címét fogjuk visszakapni.

K *Milyen típusú ICMP üzenetet küld vissza a forrásnak egy útválasztó, ha nem képes kellő sebességgel feldolgozni a tőle érkező adatokat?*

V A `Source Quench` üzenetet fogja használni.

K *Melyik címosztályhoz tartozik az az IP cím, amelynek első három bitjén az 110 minta található?*

V Ez egy C osztályú hálózat egy gépének címe.

Gyakorlatok

Alakítsuk át a következő bináris számokat decimális megfelelőjükké!

00101011	Válasz = 43
01010010	Válasz= 82
11010110	Válasz = 214
10110111	Válasz = 183
01001010	Válasz = 74
01011101	Válasz = 93
10001101	Válasz = 141
11011110	Válasz = 222

Alakítsuk át a következő decimális számokat bináris megfelelőjükké!

13	Válasz = 00001101
184	Válasz = 10111000
238	Válasz = 11101110
37	Válasz = 00100101
98	Válasz = 01100010
161	Válasz = 10100001
243	Válasz = 11110011
189	Válasz = 10111101

Hozzuk a következő bináris formában megadott IP címeket pontozott decimális formára!

11001111 00001110 00100001 01011100	Válasz = 207.14.33.92
00001010 00001101 01011001 01001101	Válasz = 10.13.89.77
10111101 10010011 01010101 01100001	Válasz = 189.147.85.97

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **ARP (*Address Resolution Protocol*)** – Az internet réteg egyik kulcsfontosságú protokollja, amely ismert IP címhez fizikai címet tud hozzárendelni a helyi hálózatban. Az ARP egy belső táblázatot (ARP cache) is fönntart ezekről a fizikai-cím-logikai-cím párokról.
- **A, B, C, D és E címosztály** – Az IP címek felosztásának hagyományos rendszere. A hálózati osztály határozza meg, hogy egy IP címben hány bit ábrázolja a hálózati azonosítót (Network ID) és hány azonosítja magát a gépet (Host ID).

- **Gépazonosító (*Host ID*)** – Az IP címnek az a része, amely a hálózat egy adott gépét azonosítja. Egy hálózat valamennyi gépének olyan IP címmel kell rendelkeznie, amelyekben egyedi a gépazonosító.
- **ICMP (*Internet Control Message Protocol*)** – Az internet réteg egyik kulcsfontosságú protokollja, amit elsősorban az útválasztók használnak arra, hogy a gépeket tájékoztassák a működésükkel kapcsolatos problémákról. Az ICMP-t használja a `ping` parancs is, amivel egy távoli gép elérhetőségét lehet ellenőrizni.
- **IP (*Internet Protocol*)** – Az internet réteg talán legfontosabb protokollja, amely a címzésért, az adatátvitelért és az útválasztásért felelős.
- **Részleges üzenetszórás (*multicast*)** – Olyan módszer, ami lehetővé teszi, hogy egy üzenetet számítógépek egy egész csoportjához juttassuk el egy helyi hálózaton belül.
- **Hálózatazonosító (*Netwrok ID*)** – Az IP címnek az a része, amely magát a hálózatot azonosítja.
- **RARP (*Reverse Address Resolution Protocol*)** – A TCP/IP csomag azon protokollja, amely egy IP címet ad vissza, ha átadunk neki egy fizikai címet. Ezt a protokollt általában olyan merevlemez nélküli munkaállomások konfigurálásához használják, amelyek a hálózati csatolójukra csatlakoztatott boot PROM segítségével hálózatról bootolnak.

5. ÓRA

Alhálózatok és a CIDR séma

Ebben az órában a következőkről lesz szó:

- Alhálózatok
- Alhálózati maszkok
- CIDR jelölések

Az alhálózatok használata (*subnetting*) lehetővé teszi, hogy az IP címzési rendszerre támaszkodva kisebb logikai egységekre, úgynevezett alhálózatokra bontsunk fel egy nagyobb fizikai hálózatot. Ebben az órában megmutatjuk, mikor van szükség erre a módszerre, milyen előnyökkel jár az alkalmazása, és hogy milyen eljárások segítségével állapíthatjuk meg a megfelelő alhálózati maszkokat.

Az óra végére a következőkkel leszünk tisztában:

- Hogyan használhatók az alhálózatok?
- Milyen előnyökkel jár egy nagyobb hálózat alhálózatokra bontása?
- Hogyan hozható összhangba az alhálózati maszkok meghatározása az üzleti folyamatokkal és igényekkel?
- Mi a lényege a hálózati osztályokon kívüli címzésnek, illetve milyen jelöléseket használ a CIDR technológia?

Alhálózatok

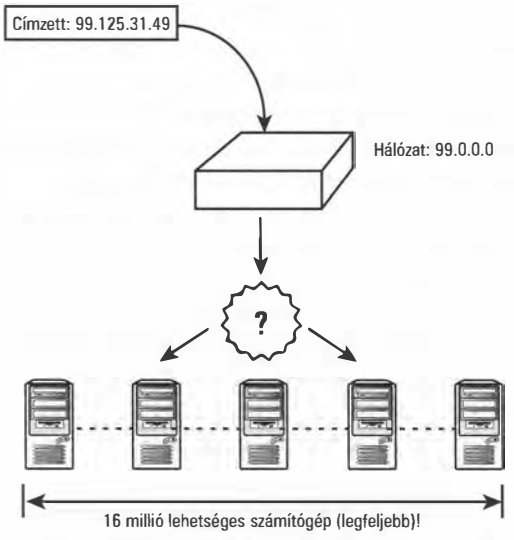
Egy IP címnek egyértelműen azonosítani kell magát a számítógépet, illetve azt a hálózatot is, amelynek az a része. Amint arról a 4. órában részletesen volt szó, a hálózati címek osztályba sorolása volt az első olyan módszer, amely alapján a hálózati eszközök különbséget tudtak tenni egy IP cím hálózatot és gépet azonosító része között. Később aztán az derült ki, hogy a hálózati osztályok rendszere túlságosan merev ahhoz, hogy a való életben kizárólag ennek a segítségével boldogulni lehessen. A valódi hálózatok egyszerűen túlságosan sokfélék, és mindenféle méretben előfordulnak. Ráadásul a többségük belül kisebb logikai egységekre tagolódik. A bajt csak tetézte a számítógépek számának megugrása, amely miatt a világ elkezdett kifutni a felhasználható osztály szintű címekből. Az internetszolgáltatóknak sürgősen szükségük volt egy olyan hajlékonyabb címzési módszerre, amellyel az osztály szintű hálózatok kisebb részekre oszthatók, az útválasztók pedig egy teljes osztálynál kisebb címtartományt is képesek kiszolgálni.

Nos, éppen erre valók az **alhálózatok használata** (*subnetting*). Segítségükkel egy nagy fizikai hálózat kisebb logikai egységekre tagolható. Az alhálózatok rendszere eredetileg igazából a hálózati osztályok rendszerén belül, és nem helyette kezdett kialakulni, s talán éppen ezért van az, hogy az alhálózatok működését megérteni ma is leginkább az A, B és C osztályú hálózatok viszonyán keresztül lehet. Ezzel együtt az internetes közösség egy idő után elvetett a címosztályokat, és egy tőlük majdnem teljesen független címzési rendszert dolgozott ki. Ez a CIDR (*Classless Internet Domain Routing*) rendszer, amely logikájában már nem igényli a címek osztályba sorolását. Ebben az órában az alhálózatok működési elvét először az osztály szintű címek rendszerén belül fogjuk bemutatni, s csak aztán lépünk tovább az ettől független CIDR rendszer bemutatására.

A hálózat felosztása

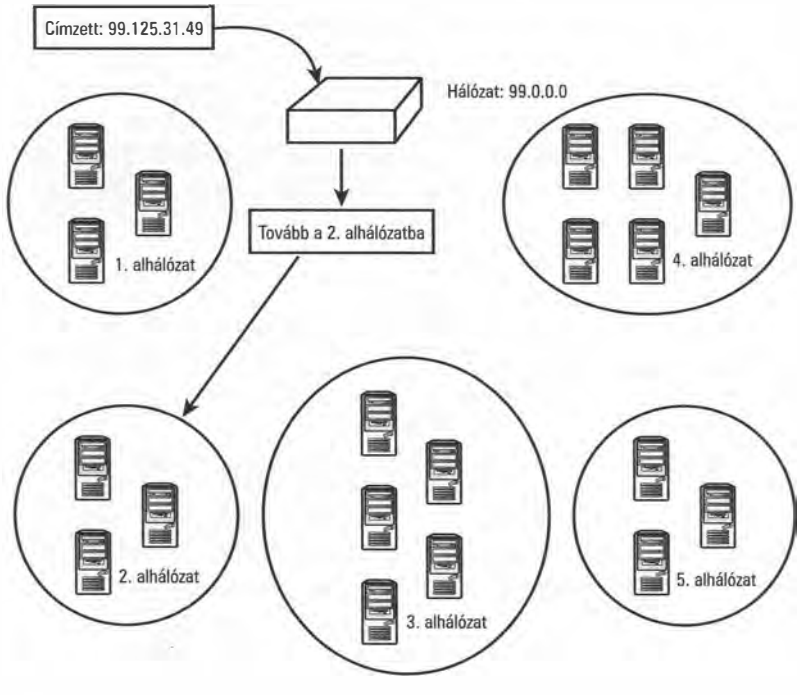
A 4. órában bemutatott címosztályok használata valamennyi útválasztó számára lehetővé teszi, hogy elkülönítse egymástól a cím hálózatot és gépet leíró részét, és így a megfelelő hálózatba továbbítsa a datagramot. Ez logikus és kényelmes rendszer ugyan, de azért van néhány hátulütője. A legfontosabb ezek közül talán az, hogy a címek A, B és C osztályba sorolása semmit nem mond az így kijelölt hálózatok belső szerkezetéről. Csak maga a hálózat az, amit meghatároz.

Az 5.1. ábrán egy A osztályú hálózat sematikus képét látjuk. Amint azt a 4. óra anyagából már tudjuk, a datagram útja a hálózat átjárójáig ki van kövezve. Hatékonyan képes rátalálni magára a 99.0.0.0 hálózatra, ám hogy ezen belül mi történjen vele, az már egy bonyolultabb kérdés. Egy A osztályú hálózatban legfeljebb 16 millió különböző cím osztható ki, tehát valószínű, hogy a hálózat sikeres felderítése után a datagramnak még mindig gépek milliói közül kell kiválasztania az igazi címzettet. Ez értelemszerűen sokkal több gép, amint ahány egy fizikai alhálózatban elfér.



5.1. ábra

Adatok továbbítása egy A osztályú hálózatba



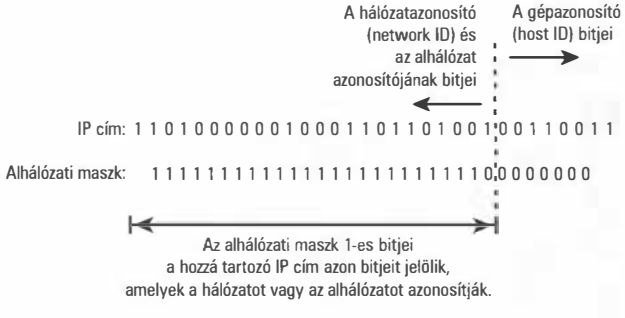
5.2. ábra

A hálózat belső felosztása a hatékony adattovábbítás végett

Ahhoz, hogy a datagram továbbítása a hálózaton belül is hatékony lehessen, a hálózatot kisebb szegmensekre kell felosztani (lásd az 5.2. ábrát). A fizikai hálózat ilyen szegmentálása egyrészt növeli a teljes hálózat hatékonyságát, másrészt lehetővé teszi, hogy a rendelkezésre álló címtartomány minél nagyobb részét használják ki. Ebben az amúgy meglehetősen gyakori helyzetben a szegmenseket összekapcsoló útválasztóknak valamiféle többletinformációra van szükségük ahhoz, hogy el tudják dönteni, merre kell továbbítaniuk egy-egy datagramot. A hálózatazonosítót (*network ID*) ők már nem használhatják, hiszen ez a hálózatba bejutott valamennyi datagramnál azonos (jelen esetben 99.0.0.0). a következő ötlet a címtér gépazonosítók alapján történő logikai felosztása lehetne, ám ha jobban belegondolunk, ez a megoldás egy 16 millió gépet tartalmazó hálózatban finoman szólva is rugalmatlan, áttekinthetetlen és végső soron kezelhetetlen lenne. A legkézenfekvőbb megoldás nyilvánvalóan az, ha a címteret a hálózatazonosító alatt osztjuk tovább úgy, hogy az útválasztók magából az IP cím ez alatti részéből tudják kitalálni, melyik alhálózati szegmens felé kell továbbítaniuk a datagramot.

Az alhálózatok rendszer tehát egy a hálózatazonosító alatti újabb logikai réteget biztosít. Az útválasztók az alhálózati cím alapján képesek a megfelelő szegmensbe továbbítani a datagramot (ez általában már egy konkrét fizikai hálózatot jelent), oda bejutva pedig az IP cím már lefordítható az ARP segítségével fizikai címmé, és a végső kézbesítést ez alapján lehet elvégezni (lásd a 4. óra anyagát).

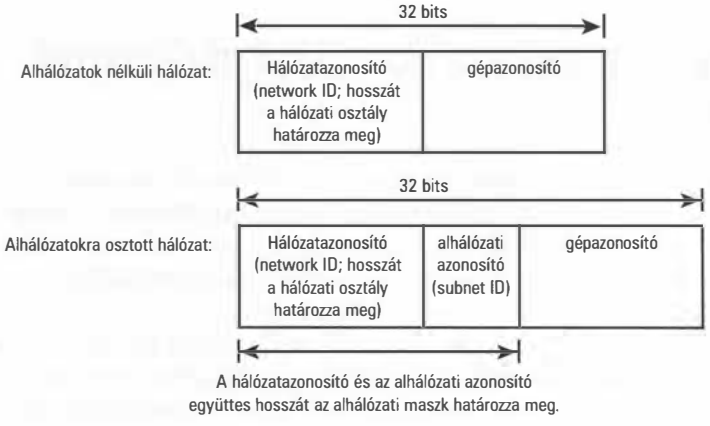
Most az olvasó bizonyára azon töpreng, honnan lesz nekünk alhálózati címünk, ha egyszer az IP cím mind a 32 bitjét elhasználtuk a hálózatazonosítóra és a gépazonosítóra. A válasz egyszerű: a TCP/IP tervezői biztosítottak számunkra egy olyan lehetőséget, amivel a gépazonosító néhány bitjét „kölcsönvehetjük", és az alhálózat azonosítására használhatjuk. Van egy **alhálózati maszknak** (*subnet mask*) nevezett paraméter, amely azt határozza meg, hogy a cím hány bitjét használjuk az alhálózat azonosítására, és mennyi marad a gépek megkülönböztetésére.



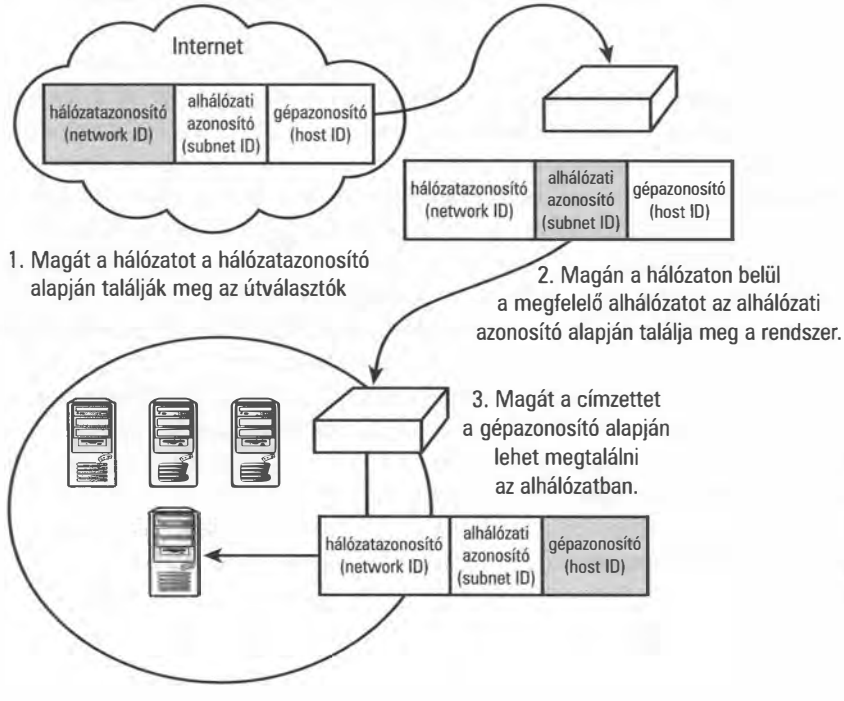
5.3. ábra
Az IP cím és az alhálózati maszk kapcsolata

Akárcsak az IP cím, az alhálózati maszk is egy 32 bites bináris szám. Ennek a számnak a bitjei úgy vannak elrendezve, hogy abból a rendszer számára kiderüljön, az IP cím pontosan melyik részét kívánjuk az alhálózat azonosítására használni. Az alhálózati maszk minden egyes bitpozíciója megfelel az IP cím ugyanazon bitpozíciójának. A cím

és a maszk ilyen megfeleltetését az 5.3. ábra mutatja. Az alhálózati maszkban minden olyan helyen 1-es bit van, ahol az IP cím megfelelő bitje a hálózatot vagy az alhálózatot hivatott azonosítani, és minden olyan helyen 0, ahol az IP cím bitje magát a gépet címzi. Az alhálózati maszk tehát egyfajta térkép, ami az IP cím értelmezésében segít a rendszernek. A címben szereplő bitek csoportosításában tehát eltérés van az alhálózatokra osztott, illetve a közönséges hálózatok között. Ezt a különbséget szemlélteti az 5.4. ábra.



5.4. ábra
A címbitek elhelyezkedése egy alhálózatokra osztott és egy közönséges hálózat esetében.



5.5. ábra
Egy bejövő datagram útja egy alhálózatokat is tartalmazó rendszerben.

Az alhálózatokra osztott hálózatokban működő útválasztók táblázataiban szerepel az az információ is, hogy mely IP címekhez mely alhálózati maszkok tartoznak. (Az átválasztásról bővebben majd a 8. órában lesz szó.) Egy datagramnak az alhálózatokkal is rendelkező hálózaton belüli útját szemlélteti az 5.5. ábra. A csomag magáig a hálózatig a cím hálózatot azonosító része (network ID) alapján jut el. Ha már bent van a célhálózatban, akkor jut szerep az alhálózati maszknak, a belső útválasztók ugyanis ez alapján fogják eljuttatni a megfelelő belső szegmensbe. Ha ide is sikeresen megérkezett, akkor a végső kézbesítés a gépazonosító (host ID), illetve a fizikai cím alapján történik.

Az alhálózati maszk átalakítása pontokkal elválasztott decimális formává

A hálózati adminisztrátor általában az IP cím kiosztásával egy időben minden géphez hozzárendel egy alhálózati maszkot is. Ez az érték tehát része a TCP/IP beállításainak. Ha olyan gépről van szó, amely az IP címét DHCP-n keresztül kapja (lásd majd a 12. órában), akkor az a rá vonatkozó alhálózati maszkot is ilyen módon fogja megkapni.

Az alhálózati maszkok értékének meghatározásakor körültekintően kell eljárnunk, mert ezeknek a számoknak pontosan tükrözniük kell a hálózat belső logikai felépítését. Az azonos alhálózathoz tartozó gépeknek természetesen azonos alhálózati maszkkal kell rendelkezniük. Ugyanazon okokból kifolyólag, amiért az IP címet sem annak bináris formájában használjuk, az alhálózati maszkot is át szokás alakítani annak pontokkal elválasztott decimális formájává.

Amint az előző szakaszban már tisztáztuk, az alhálózati maszk ugyanúgy 32 bites bináris szám, mint az IP cím, tehát az átalakításánál is ugyanazt az algoritmust követhetjük, amit a 4. órában már ismertettünk. Annyit talán érdemes hozzátenni a dologhoz, hogy az alhálózati maszk sokkal kisebb változatosságot mutat, mint az IP cím, így az átalakítása is sokkal könnyebb. Az alhálózati maszk azon pozícióiban, amelyek az IP címben a hálózatazonosítónak felelnek meg, 1-eseket találunk, míg a gépazonosítót 0-s bitek jelzik. Ez egyben azt is jelenti, hogy (néhány meglehetősen vad kivételtől eltekintve) az egyesek mindig a maszk bal oldalán sorakoznak, míg a nullák a jobb oldalát foglalják el. A decimális formában minden olyan oktetnek, amely csupa egyesből áll (11111111) a 255 felel meg, míg a csupa nullából álló oktet (00000000) tízes számrendszerbeli megfelelője is 0. a talán leggyakrabban használt alhálózati maszk a következő:

11111111111111111111111100000000

Ennek a pontokkal elválasztott decimális formája a fentieknek megfelelően a következő: 255.255.255.0.

Teljesen hasonlóan a

11111111111111110000000000000000

alhálózati maszk decimális megfelelője a 255.255.0.0.

Látható tehát, hogy olyan alhálózati maszk esetében, amely éppen két oktet határán választja kettő a címet, egészen egyszerűen meghatározható a megfelelő decimális alak. No de mi a helyzet azokkal a maszkokkal, amelyek nem ennyire szabályosak? A helyzet ebben az esetben sem sokkal bonyolultabb, hiszen csak annyi a dolgunk, hogy kiemeljük a bináris alakból az 1-eseket és 0-kat vegyesen tartalmazó bitnyolcast és azt a már ismert algoritmus segítségével decimális alakra hozzuk.

Egy alhálózati maszk pontokkal elválasztott decimális alakjának meghatározására tehát a következő általános algoritmus adható:

1. Először is osszuk fel a maszk bináris alakját bitnyolcasokra úgy, hogy minden nyolcadik 1-es vagy nulla után beszúrunk egy pontot. Íme egy példa:
   11111111.11111111.11110000.00000000
2. Minden csupa egyesekből álló oktet helyett írjunk 255-öt, minden csupa nullából álló helyett pedig egyetlen 0-t.
3. Az egyeseket és nullákat egyaránt tartalmazó oktet átalakítására használjuk a 4. órában megismert eljárást (lásd a 4.5. ábrát), amit röviden úgy foglalhatunk össze, hogy össze kell adni azon helyiértékeket, ahol 1-es látunk.
4. Végül írjuk le a megfelelő sorrendben az egyes oktetek decimális formáját pontokkal elválasztva. Esetünkben ez a következő lesz:
   255.255.240.0

Az esetek túlnyomó többségében ezt az értéket minden számítógépen meg kell adnunk, mégpedig a TCP/IP rendszer első beállításakor.

Munka alhálózatokkal

Hogy a címben a hálózatazonosító (*network ID*) után következő bitek közül hányat használunk az alhálózat meghatározására, azt az alhálózati maszk értéke határozza meg. Ez azt is jelenti, hogy az alhálózati azonosító (*subnet ID*) hossza változó lehet, attól függően, miként kívánjuk strukturálni a hálózatot. Ahogy nő az alhálózati azonosító hossza, úgy csökken a gépazonosítóé (*host ID*), ami azt jelenti, hogy ha hálózatunkban sok alhálózatot jelölünk ki, akkor ezekben viszonylag kevés gép lehet. A dolog persze fordítva. Is igaz, vagyis ha csak kevés alhálózatot üzemeltetünk, akkor ezek megkülönböztetésére kevés bit is elegendő, így több gép tartozhat az egyes logikai szegmensekhez.

Érdemes megjegyezni, hogy az alhálózati azonosító lehetséges hosszát a hálózati osztály is meghatározza, hiszen a különböző osztályokba tartozó címek esetében eltérő számú bittel gazdálkodhatunk. A következő maszk például

11111111111111111110000000000000

19 bitet rendel a hálózatazonosítóhoz és az alhálózati azonosítóhoz együttesen. Ha a cím, amire ez a maszk vonatkozik egy B osztályú (Class B) cím , amely mint tudjuk 16 bitet rendel a hálózatazonosítóhoz, akkor mindössze 3 bit áll rendelkezésünkre az alhálózati azonosító ábrázolásához. Ha ellenben a fenti maszkot egy A osztályú címtartományban használjuk, akkor 11 bitünk van az alhálózatok megkülönböztetésére.

Az alhálózati azonosítók (és így az alhálózati maszkok) hozzárendelésének módja természetesen a hálózat felépítésétől függ. A legjobb megoldás az, ha először töviről hegyire megtervezzük hálózatunk logika szerkezetét, kijelöljük az összes önálló szegmenst, aztán ezekhez egyenként hozzárendelünk egy-egy alhálózati azonosítót. Ha erre lehetőségünk van, tartalékoljunk némi helyet az új alhálózatoknak is, mert szinte biztos, hogy lesznek ilyenek.

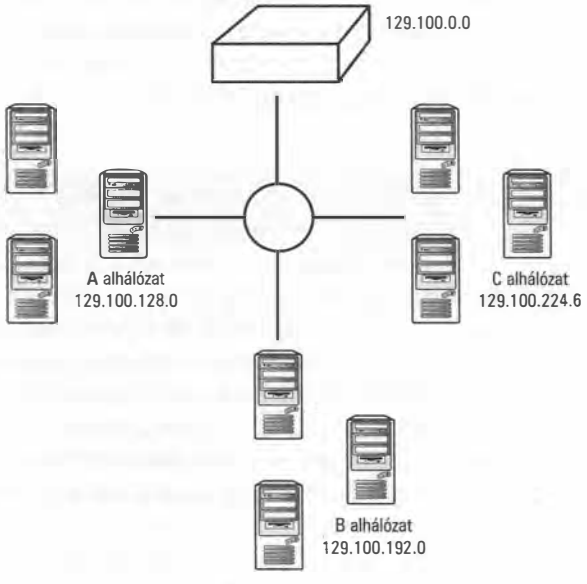
Az alhálózatok kijelölésének legegyszerűbb példája talán az, amikor egy B osztályú (Class B) hálózatban a teljes harmadik oktetet (vagy ha úgy jobban tetszik a decimális cím harmadik tagját) jelöljük ki alhálózati azonosítónak. Az 5.6. ábrán vázoltuk azt a helyzetet, amikor a 129.100.0.0 hálózatot bontjuk négy alhálózatra. A hálózat valamennyi tagján a 255.255.255.0 alhálózati maszkot állítjuk be, ami azt jelenti, hogy a hálózatazonosító és az alhálózati azonosító együttesen az IP cím első három oktetjét foglalják el. Mivel maga a hálózat egy B osztályú címtartománnyal rendelkezik (lásd a 4. órát), a hálózatazonosító az első 16 bitet foglalja el. Az 5.6. ábrán bemutatott hálózat tehát a következő paraméterekkel rendelkezik:

Hálózatazonosító: 129.100.0.0
Alhálózati azonosító: 0.0.128.0

A gépazonosító (*host ID*) a szabályok szerint nem állhat sem csupa egyesből, sem csupa nullából, vagyis az 5.6. ábrán bemutatott hálózatban összesen 254 alhálózat jelölhető ki, és ezek mindegyike 254 számítógépet tartalmazhat. Ez a megoldás kiválóan működik egészen addig, amíg nincs szükség 254 címnél többre egyetlen szegmensen belül sem, no és feltéve persze, hogy birtokunkban van egy teljes B osztályú címtartomány (amit mostanában egyre nehezebb találni).

Gyakran nincs arra lehetőségünk, hogy a címekből egy teljes bitnyolcast áldozzunk az alhálózati azonosítónak. Ha például egy C osztályú hálózatban ezt tennénk, akkor egyetlen bitünk sem maradna a gépek azonosítására. Mi több, még az is könnyen

előfordulhat, hogy egy B osztályú hálózatban sem használhatunk el 8 bitet erre a célra, hiszen ha bármely szegmensben 254-nél több címre van szükség, akkor máris borul a terv. A nem oktethatárra eső alhálózati maszkot igen egyszerűen szemléltethetjük bináris alakban, de kissé zavarossá válik a dolog, ha áttérünk a decimális formára.



5.6. ábra
Egy alhálózatokra osztott B osztályú hálózat.

Képzeljünk el egy C osztályú hálózatot, amit öt kisebb alhálózatra szeretnénk fölosztani. A C címosztály definíciója szerint a hálózatazonosító után marad még nyolc bitünk, amelyeket az alhálózat és az egyes gépek azonosítására használhatunk. Ezzel kell jól gazdálkodnunk. Tegyük fel, hogy három bitet szeretnénk elhasználni az alhálózati azonosító megadására. Ez azt jelenti, hogy az alhálózati maszk a következőképpen fog kinézni:

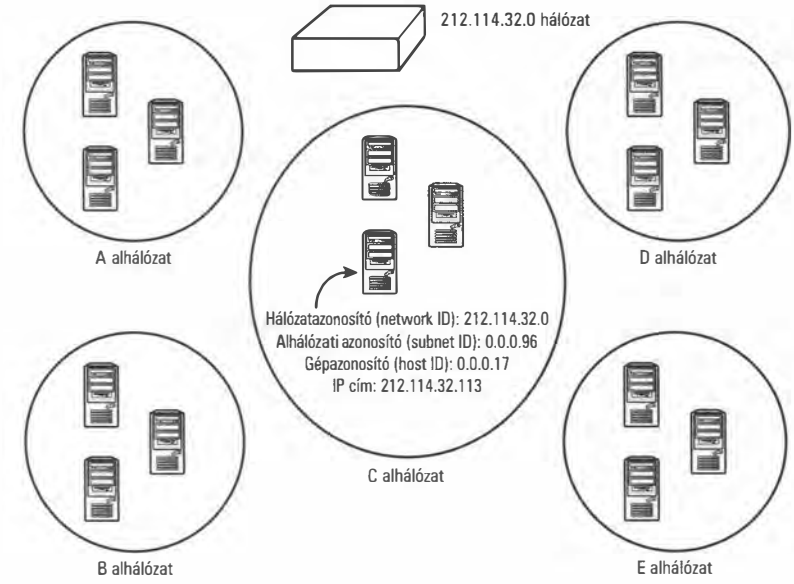
11111111111111111111111111100000

Ebben a fölállásban a gépazonosító megadására még 5 bitünk van. Az alhálózati azonosító számára fönntartott három biten összesen nyolcféle bitmintát tudunk megadni. Amint azt korábban említettük, a csupa egyesből, illetve csupa nullából álló azonosító nem használható, tehát maradt összesen hat lehetőségünk. Ez öt alhálózat kijelöléséhez mindenképpen elegendő. (Az igazsághoz hozzátartozik, hogy számos útválasztó valójában tudja kezelni a csupa egyesből és csupa nullából álló azonosítókat is, csak ez nem egy szabványos lehetőség.) A gépek azonosítására megmaradt 5 biten 32-féle bitminta ábrázolható, ám ha megint kizárjuk a csupa egyesből és a csupa nullából álló kombinációt, akkor összesen 30 gépünk lehet egy-egy szegmensben.

Ha az imént megadott alhálózati maszkot annak decimális formájára szeretnénk hozni, nincs más teendőnk, mint követni az imént már ismertetett eljárást:

1. Tegyünk pontokat a bitnyolcasok határára:
   11111111.11111111.11111111.11100000
2. Írjunk le 255-öt minden csupa egyesből álló oktet helyett. Az egyetlen egyeseket és nullákat egyaránt tartalmazó bitnyolcast számítsuk át tízes számrendszerbe:
   128+64+32=224
3. Írjuk le egymás után pontokkal elválasztva a kapott négy számot:
   255.255.255.224

Tegyük fel, hogy elkezdjük benépesíteni ezt a bizonyos hálózatot gépekkel, és mindegyikhez hozzárendelünk egy egyedi címet (5.7. ábra). Mivel egy C osztályú címtartományt kezelünk, a decimális címek első három tagja mindenütt azonos lesz. A negyedik tag kiszámításánál nincs más dolgunk, mint beírni egymás után az alhálózati maszk és a gépazonosító bitjeit a megfelelő pozícióba. Az 5.7. ábrán látható C alhálózat esetében például magát az alhálózatot a 011 bitminta azonosítja. Mivel ezek a bitek az utolsó oktat bal felére esnek, az alhálózati azonosító valójában a 01100000 bináris számnak felel meg, ami decimális alakban 96. Ha a gépazonosító decimális 17, vagyis az 10001 bitminta, akkor a cím negyedik bitnyolcasa teljes szépségében a következőképpen fest: 01110001, ami decimális alakban 113. Ennek a bizonyos számítógépnek a teljes IP címe tehát 212.114.32.113 lesz.



5.7. ábra

Egy alhálózatokra bontott C osztályú hálózat.

A pontozott decimális alakban megadott alhálózati maszkoknak megfelelő bitmintákat az 5.1. Táblázatban foglaltuk össze. Ebben valamennyi érvényes alhálózati bitminta megtalálható. A „Leírás" mezőben feltüntettük azoknak az egyes biteknek a számát, amelyek a hálózati osztály által meghatározott alapértelmezett maszkon felül rendelkezésünkre állnak. Az adott címosztály esetén ennyi bitünk van tehát az alhálózat azonosítására. Egy A osztályú címhez tartozó alapértelmezett alhálózati maszkban például nyolc egyes van. Az ehhez az osztályhoz tartozó sorok között a táblázatban van egy olyan, amelyben a maszkbitek száma kettő. Ez a sor azt jelenti, hogy ebben a felállásban nyolc plusz kettő, vagyis összesen 10 egyes bitünk van az alhálózati maszkban.

5.1. Táblázat *Pontozott decimális formában megadott alhálózati maszkok és a hozzájuk tartozó bitminták*

Leírás	Pontozott decimális forma	Bitminta
A osztályú hálózatok (Class A)		
Alapértelmezett maszk	255.0.0.0	11111111 00000000 00000000 00000000
1 alhálózati bit	255.128.0.0	11111111 10000000 00000000 00000000
2 alhálózati bit	255.192.0.0	11111111 11000000 00000000 00000000
3 alhálózati bit	255.224.0.0	11111111 11100000 00000000 00000000
4 alhálózati bit	255.240.0.0	11111111 11110000 00000000 00000000
5 alhálózati bit	255.248.0.0	11111111 11111000 00000000 00000000
6 alhálózati bit	255.252.0.0	11111111 11111100 00000000 00000000
7 alhálózati bit	255.254.0.0	11111111 11111110 00000000 00000000
8 alhálózati bit	255.255.0.0	11111111 11111111 00000000 00000000
9 alhálózati bit	255.255.128.0	11111111 11111111 10000000 00000000
10 alhálózati bit	255.255.192.0	11111111 11111111 11000000 00000000
11 alhálózati bit	255.255.224.0	11111111 11111111 11100000 00000000
12 alhálózati bit	255.255.240.0	11111111 11111111 11110000 00000000
13 alhálózati bit	255.255.248 0	11111111 11111111 11111000 00000000
14 alhálózati bit	255.255.252.0	11111111 11111111 11111100 00000000
15 alhálózati bit	255.255.254.0	11111111 11111111 11111110 00000000
16 alhálózati bit	255.255.255.0	11111111 11111111 11111111 00000000
17 alhálózati bit	255.255.255.128	11111111 11111111 11111111 10000000
18 alhálózati bit	255.255.255.192	11111111 11111111 11111111 11000000
19 alhálózati bit	255.255.255.224	11111111 11111111 11111111 11100000
20 alhálózati bit	255.255.255.240	11111111 11111111 11111111 11110000
21 alhálózati bit	255.255.255.248	11111111 11111111 11111111 11111000
22 alhálózati bit	255.255.255.252	11111111 11111111 11111111 11111100
B osztályú hálózatok (Class B)		
Alapértelmezett maszk	255.255.0.0	11111111 11111111 00000000 00000000
1 alhálózati bit	255.255.128.0	11111111 11111111 10000000 00000000
2 alhálózati bit	255.255.192.0	11111111 11111111 11000000 00000000
3 alhálózati bit	255.255.224.0	11111111 11111111 11100000 00000000

5.1. Táblázat *Pontozott decimális formában megadott alhálózati maszkok és a hozzájuk tartozó bitminták (folytatás)*

Leírás	Pontozott decimális forma	Bitminta
4 alhálózati bit	255.255.240.0	11111111 11111111 11110000 00000000
5 alhálózati bit	255.255.248.0	11111111 11111111 11111000 00000000
6 alhálózati bit	255.255.252.0	11111111 11111111 11111100 00000000
7 alhálózati bit	255.255.254.0	11111111 11111111 11111110 00000000
8 alhálózati bit	255.255.255.0	11111111 11111111 11111111 00000000
9 alhálózati bit	255.255.255.128	11111111 11111111 11111111 10000000
10 alhálózati bit	255.255.255.192	11111111 11111111 11111111 11000000
11 alhálózati bit	255.255.255.224	11111111 11111111 11111111 11100000
12 alhálózati bit	255.255.255.240	11111111 11111111 11111111 11110000
13 alhálózati bit	255.255.255.248	11111111 11111111 11111111 11111000
14 alhálózati bit	255.255.255.252	11111111 11111111 11111111 11111100
C osztályú hálózatok (Class C)		
Alapértelmezett maszk	255.255.255.0	11111111 11111111 11111111 00000000
1 alhálózati bit	255.255.255.128	11111111 11111111 11111111 10000000
2 alhálózati bit	255.255.255.192	11111111 11111111 11111111 11000000
3 alhálózati bit	255.255.255.224	11111111 11111111 11111111 11100000
4 alhálózati bit	255.255.255.240	11111111 11111111 11111111 11110000
5 alhálózati bit	255.255.255.248	11111111 11111111 11111111 11111000
6 alhálózati bit	255.255.255.252	11111111 11111111 11111111 11111100

Az 5.1. táblázatban bemutatott paraméterkombinációk között akad néhány olyan, amely elvileg lehetséges ugyan, de semmiféle gyakorlati haszna nincs. A gyakorlatban például semmit nem tudunk kezdeni egy olyan C osztályú hálózattal, amelyben hat bit azonosítja az alhálózatot, és csak kettő marad a gépek azonosítására. A két biten ábrázolható négy bitkombináció közül kiesik a csupa egyes (11), mert ez az üzenetszórási cím, és a csupa nullát (00) sem szokták használni. Maradt tehát két kombináció, ami azt jelenti, hogy alhálózatonként mindössze két gépünk lehet, ennek pedig gyakorlati jelentőséget tulajdonítani meglehetősen nehéz.

A CIDR (Classless Internet Domain Routing) címzési séma

A osztályú címtartományhoz hozzáférni már régóta nem lehet, sőt, a világ hálózatai hamarosan elhasználják a B osztályú címtartományokat is. C osztályba tartozó címtartományok egyelőre még igényelhetők, ezekkel azonban az a gond, hogy bennük legfeljebb 254 cím osztható ki, ami elenyésző ahhoz képest, ahány ügyfele egy komolyabb internet szolgáltatónak (*Internet Service Provider; ISP*) van. Márpedig külön hálózatként kezelni olyan C osztályú hálózatokat, amelyek a valóságban egyetlen rendszert alkotnak nem egyéb, mint az útválasztási táblák szükségtelen túlbonyolítása.

Amint azt már korábban is említettük, a hálózati osztályok rendszere meglehetősen merev, és még ahhoz is extra szolgáltatásra (az alhálózati maszkra) van szükség, ha egy adott hálózatot szeretnénk belső logikai szerkezettel felruházni. A CIDR (*Classless Internet Domain Routing*) egy olyan, sokkal inkább a valós igényekhez alkalmazkodó címzési séma, mellyel az útválasztási táblákban egész címblokkok adhatók meg. A CIDR rendszer működése nem támaszkodik egy előre meghatározott hosszúságú (8, 16 vagy 24 bites) hálózatazonosító használatára. Helyette egyetlen szám, az úgynevezett CIDR előtag (*CIDR prefix*) adja meg a hálózat azonosítására használt bitek számát. Ezt az előtagot szokás változó hosszúságú alhálózati maszknak (*Variable Length Subnet Mask; VLSM*) is nevezni. Az előtag a címtartományon belül bárhova eshet, így egy könnyen átlátható és kényelmes módszert és egyben jelölésrendszert biztosít a hálózat és a számítógép azonosítására szolgáló bitek elkülönítésére. A CIDR jelölésben a cím után egy perjel következi, utána pedig egy tízes számrendszerbeli szám, amely azt mutatja meg, hogy az adott címből hány bit azonosítja a hálózatot. A 205.123.196.183/25 CIDR formátumú cím például azt jelenti, hogy a cím első 25 bitje a hálózatazonosító. Ez a korábbi rendszerben gondolkodva a 255.255.255.128 alhálózati maszknak felel meg.

Összességében tehát a CIDR előtag tulajdonképpen azt határozza meg, hogy az adott hálózatban található gépek IP címeiből az első hány bit azonos. Ennek a módszernek az egyik óriási előnye az, hogy segítségével nem csak felosztani lehet egy hálózatot kisebb logikai egységekre, hanem össze is lehet vonni több kisebb hálózatot egyetlen ilyen egységgé. Ha például egy ISP meg tud szerezni több egymást követő C osztályú címtartományt, akkor a CIDR segítségével ezeket egyetlen logikai címtartománnyá olvaszthatja össze. Az IPv4 életét tulajdonképpen a CIDR címzési rendszer mentette meg (átmenetileg), mivel nagyban egyszerűsítette az útválasztási táblák kezelését annak ellenére is, hogy a kiosztható címek vészesen fogyatkoztak. Ha egy ISP több, egymás után következő C osztályú címtartománnyal rendelkezett, az útválasztási táblában akkor is csak egy CIDR bejegyzésre volt szükség ennek a leírásához. Az ilyen esetekben tehát a CIDR nem elválaszt, hanem összeköt hálózatokat, ezért is hívják néha **szuperhálózati maszknak** (*supernet masknak*). Segítségével egy internetszolgáltatónak kiosztható például az összes olyan C osztályú hálózat, amelyek a 204.21.128.0-tól (11001100000101011000000000000000) a 204.21.255.255-ig (11001100000101011111111111111111) terjednek.

Ez az összefüggő címtartomány az eredeti, osztályokra alapozott címzési modellben semmiféle osztálynak nem felelt meg, mivel azonban a fenti tartományba eső címekben az első tizenhét bit azonos, a CIDR segítségével egyben kezelhetők. A nekik megfelelő szuperhálózati maszk az 11111111111111111000000000000000 lesz, amely pontozott decimális formában 255.255.128.0-ként írható le.

Magát a címblokkot úgy szokás megadni, hogy leírjuk a benne előforduló legalacsonyabb címet, majd utána szuperhálózati maszkban szereplő egyesek számát. Fenti példánknál maradva tehát a kérdéses összevont címtartomány a következő CIDR bejegyzéssel írható le: 204.21.128.0/17. Ez a bejegyzés illeszkedik az összes olyan címre, amelyek első 17 bitje megegyezik a 204.21.128.0 IP cím első 17 bitjével.

Összefoglalás

Az alhálózatok használata egyfajta közbülső réteget biztosít az IP címek rendszerében azáltal, hogy lehetőséget ad az azonos hálózatazonosítóval rendelkező címek csoportosítására. Az alhálózatok alkalmazása általánosnak tekinthető minden olyan helyen, ahol a hálózati infrastruktúra belső útválasztók révén több fizikai szegmensre tagolódik.

Létezik egy viszonylag újabb keletű címzési rendszer is, a CIDR (*Classless Internet Domain Routing*), amely egy a 4. órában megismert osztályoktól független, s így sokkal hajlékonyabb módszert biztosít a címtér felosztására.

Kérdések és válaszok

K *Hány bites az alhálózati azonosító egy olyan B osztályú hálózatban, amelyben a 255.255.0.0 alhálózati maszkot használjuk?*

V Nulla, vagyis tulajdonképpen nincs alhálózati azonosító a címben. A 255.255.0.0 maszk a B osztályú hálózatok alapértelmezett alhálózati maszkja, és azt jelenti, hogy mind a 16 bitet hálózatazonosítóként használjuk, a gépazonosítóból pedig nem veszünk el további biteket az alhálózati azonosító kedvéért. Egy ilyen hálózat egyetlen logikai egységet alkot, nincsenek benne alhálózatok.

K *Egy hálózati adminisztrátor kiszámolta, hogy a hálózatok és alhálózatok azonosítására összesen 21 bitre lesz szüksége. Milyen alhálózati maszkot használjon?*

V 21 maszkbit a következő bitmintának felel meg: 11111111111111111111100000000000. Ebben két oktet csupa egyesekből áll, vagyis a maszk két 255-össel fog kezdődni. A harmadik oktet elején öt darab egyes bit van, amit a következőképpen hozhatunk decimális alakra: 128+64+32+16+8=248. A maszk tehát 255.255.248.0.

K *Cégünknek van egy C osztályú címtartománya. A vállalat 10 telephelyen működik, amelyek mindegyikén 12 ember dolgozik. Milyen alhálózati maszk vagy maszkok használhatók ebben a hálózatban, ha azt akarjuk, hogy minden dolgozónak legyen saját munkaállomása?*

V A 255.255.255.240 alhálózati maszk összesen négy bitet hagy meg a gépek azonosítására, ez pedig elegendő arra, hogy minden telephelyen minden gépnek tudjunk címet adni.

K *Bill egy A osztályú címtartománnyal rendelkezik és 3 bitet szeretne használni az alhálózatok azonosítására. Milyen alhálózati maszkot használjon?*

V Az A címosztály azt jelenti, hogy a hálózatot a címek első 8 bitje azonosítja. A maszk első oktetje tehát decimális alakban biztosan 255 lesz. A második oktet első három bitje fogja azonosítani az alhálózatot. Ezek értéke decimálisan 128+64+32=224. Az alkalmazandó alhűlózati maszk tehát 255.224.0.0.

K *Milyen IP címeknek felel meg a 212.100.192.0/20 CIDR tartomány?*

V A /20 szuperhálózati paraméter (supernet parameter) azt jelenti, hogy az IP címek első 20 bitje azonos lesz, és csak a többi fog változni. Az első ilyen cím bináris alakban a következő:

11010100.01100100.11000000.00000000

A tartomány legmagasabb címében az első 20 bit nyilván azonos lesz a fentivel, hiszen csak a többi változhat, a többi helyen pedig csupa egyesnek kell lennie, hiszen így kapjuk a legnagyobb bináris számot. A felső határ tehát (csupa egyes az iménti csupa nulla helyett):

11010100.01100100.11001111.11111111

Decimális alakban a megadott címtartomány tehát a 212.100.192.0-tól a 212.100.207.255-ig terjed.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **CIDR** – A Classless Internet Domain Routing rövidítése. Ez egy olyan címzési eljárás, amely lehetővé teszi, hogy IP címek egy egybefüggő blokkját egyetlen logikai egységként kezeljük.
- **Alhálózat (*subnet*)** – A TCP/IP hálózatazonosítója által meghatározott címtartomány felosztása kisebb részekre.
- **Alhálózati maszk (*subnet mask*)** – 32 bites bináris szám, amely meghatározza, hogy az IP cím hány bitjét kell az alhálózat címének tekinteni.
- **Szuperhálózati maszk (*supernet mask*)** – 32 bites bináris szám, melynek segítségével egymást követő hálózati azonosítók egyetlen logikai egységgé vonhatók össze.

6. ÓRA

A szállítási réteg

Ebben az órában a következőkről lesz szó:

- Kapcsolatközpontú (*connection-oriented*) és kapcsolat nélküli (*connectionless*) protokollok
- Kapuk (*port*) és foglalatok (*socket*)
- A TCP protokoll
- Az UDP protokoll

A szállítási réteg (*transport layer*) egyrészt programozói interfészt nyújt a hálózati alkalmazások számára, másrészt igény szerint hibaellenőrzést, folyamatszabályozást és a hálózati átvitel ellenőrzését is képes biztosítani. Ebben az órában áttekintjük a szállítási réteg működésével kapcsolatos legfontosabb elveket, illetve megismerkedünk a TCP és az UDP protokollal.

Az óra végére a következőkkel leszünk tisztában:

- Mik a szállítási réteg alapvető feladatai?
- Mi az alapvető különbség egy kapcsolatközpontú (*connection-oriented*) és egy kapcsolatmentes (*connectionless*) protokoll között?
- Hogyan biztosít programozási felületet a kapukon (*port*) és foglalatokon (*socket*) keresztül a szállítási réteg a hálózati alkalmazások számára?
- Mik a főbb különbségek a TCP és az UDP protokoll között?
- Milyen mezőket tartalmaz egy TCP fejléc?
- Hogyan nyit meg és zár le a TCP egy kapcsolatot?
- Hogyan rendezi sorba a TCP az átvinni kívánt adatcsomagokat, és hogyan igazolja vissza azok megérkezését?
- Mi a rendeltetése annak a négy mezőnek, amelyekből egy UDP csomag fejléce áll?

A szállítási réteg funkcióinak áttekintése

Amint azt a 4. és 5. órából már tudjuk, az internet réteg kiváló megoldásokat nyújt a címzés és útválasztás megvalósításához, ám ez a megoldandó feladatnak csak egy része. Ezt nyilván a TCP/IP fejlesztői is tudták, ezért terveztek még egy réteget az internet réteg fölé, amely az IP-vel kommunikálva további szolgáltatásokat nyújt a hálózati alkalmazásoknak. Ez a szállítási réteg, amelynek protokolljai a következőkre hivatottak:

- **Interfészt biztosítanak a hálózati alkalmazások számára** – Ez röviden azt jelenti, hogy a szállítási réteg biztosítja a hálózati alkalmazások számára azt a szolgáltatást, vagy felületet, amelyen keresztül azok a hálózathoz hozzáférhetnek. A köztes rétegre e tekintetben azért volt szükség, mert a tervezők nem csupán azt szerették volna megoldani, hogy a forgalmazott adatokat egy bizonyos gépnek lehessen címezni, hanem azt is, hogy az adott gépen futó programok közül az a másik hálózati alkalmazás is kijelölhető legyen, amelynek az adatokat fogadnia kell.
- **Megoldást adnak az adatok multiplexelésére és visszaalakítására (demultiplexing)** – A multiplexelés ebben az esetben azt jelenti, hogy egy számítógép egyszerre több másiktól képes adatokat fogadni, és azokat szétválogatva a megfelelő alkalmazáshoz tudja továbbítani. Másként fogalmazva a szállítási rétegnek képesnek kell lennie arra, hogy egyszerre több hálózati alkalmazással kommunikáljon, és ennek megfelelően szabályozza az internet réteg felé irányuló adatforgalmat. A fogadó oldalon szintén a szállítási rétegnek kell tudnia szétválogatni az internet rétegtől kapott adatokat, és azokat a megfelelő hálózati alkalmazáshoz irányítani. Ezt a funkciót nevezik általánosan demultiplexelésnek vagy visszakódolásnak, lényege pedig jelen esetben az, hogy a küldő és a fogadó egyszerre több, a hálózatot használó alkalmazást is képes futtatni. Például egyszerre működhet rajta egy webböngésző, egy levelezőkliens és egy fájlmegosztás. A multiplexelés/

demultiplexelés képességének van egy másik előnye is. Segítségével ugyanis nem csak több alkalmazás működhet egy időben, hanem egy adott alkalmazás több távoli géppel is tarthat kapcsolatot egyszerre.

- **Hibaellenőrzést, folyamatszabályozást és ellenőrzést végeznek** – A teljes protokollrendszernek szüksége van egy olyan elemre, amely átfogó módon ellenőrzi a továbbított adatok helyességét, illetve azt, hogy egyáltalán megérkeztek-e a rendeltetési helyükre. Ezt a feladatot a szállítási réteg látja el.

A felsoroltak közül talán az utolsó pont az, ami a legkevésbé egyértelműen írja el, mi is az adott dologgal kapcsolatban a szállítási réteg feladat. Itt ugyanis egyfajta minőségbiztosításról van szó, ami mindig a szolgáltatás minősége és annak ára közti mérlegelést jelent. Egy finoman kidolgozott biztonsági szabályokkal ellátott rendszer gyakorlatilag száz százalékos valószínűséggel képes biztosítani az adatok hibátlan átvitelét, ennek azonban ára lesz. Lassabban fog működni az átvitel, nagyobb feldolgozási teljesítményt igényel a rendszer működtetése, illetve az állandó visszaigazolások és ellenőrzések miatt nagyobb lesz az összesített hálózati forgalom. Számos olyan alkalmazási terület létezik, ahol ennek a megvalósítás egyszerűen nem éri meg. Éppen ezért a szállítási réteg két különböző adatátviteli sémát is biztosít a hálózathoz való hozzáférésre. Mindkettőnél hasonló a használható programozási felület, mindkettőnél támaszkodhatunk a rendszer által biztosított multiplexelés/demultiplexelés képességére, de van egy pont, amiben a két séma erősen eltér egymástól, mégpedig az átvitel minőségbiztosításának megközelítése. A szállítási réteg a következő két protokollt bocsátja rendelkezésünkre:

- **TCP (Transport Control Protocol)** – A TCP finoman kidolgozott hibaellenőrzési folyamatszabályozási szolgáltatásokat nyújt, amelyek révén garantálni tudja az adatok helyes és maradéktalan átvitelét. A TCP kapcsolatközpontú (*connection-oriented*) protokoll.
- **UDP (User Datagram Protocol)** – Az UDP kizárólag a legalapvetőbb hibaellenőrzési szolgáltatásokat nyújtja csak, és alapvetően akkor használják, amikor a TCP kifinomult ellenőrzési funkcióira nincs szükség. Az UDP kapcsolatmentes (*connectionless*) protokoll.

A kapcsolatközpontú és kapcsolatmentes protokollok közti különbségről, illetve a TCP és az UDP működéséről az óra anyagában még részletesen esik majd szó.

A TCP/IP szállítási rétege (Transport Layer) funkcióit tekintve teljesen megfelel az OSI modell ugyanilyen nevű rétegének. Az OSI modell szállítási rétegét szokás 4. rétegként (Layer 4) is említeni.

A szállítási réteggel kapcsolatos fogalmak

Mielőtt belekezdenénk a TCP és az UDP protokollok részletes tárgyalásába, talán nem árt tisztázni egyes a szállítási réteggel kapcsolatos fogalmak pontos jelentését. A következőkkel kell tisztában lennünk:

- Kapcsolatközpontú és kapcsolatmentes protokollok
- Kapuk (port) és foglalatok (socket)
- Multiplexelés és demultiplexelés

Ezek azok az alapvető koncepciók és fogalmak, amelyek ismerete mindenképpen szükséges ahhoz, hogy megérthessük a szállítási réteg működését. Éppen ezért a következő szakaszokban egyenként sorr avesszük őket.

Kapcsolatközpontú és kapcsolatmentes protokollok

Annak érdekében, hogy minden szituációban a megfelelő szintű minőségbiztosítást legyenek képesek biztosítani, a TCP/IP fejlesztői eleve két koncepcióval álltak elő, melyek alapján két különböző logikával működő ősprotokoll készült el.

- A kapcsolatközpontú protokoll (*connection-oriented protocol*) működésének lényege, hogy logikai kapcsolatot létesít a kommunikáló felek között, majd ezt a kommunikáció teljes időtartama alatt fönntartja, illetve követi az állapotát. Ez a gyakorlatban azt jelenti, hogy a küldő gép minden egyes csomagról visszaigazolást kap, ezeket a visszaigazolásokat maga is nyilvántartja, így folyamatosan információval rendelkezik arról, hogy az átvinni kívánt adatok mely része érkezett meg sikeresen, és mely csomagokat kell újra elküldeni. Az átvitel végén a küldő és a fogadó a megadott szabályok szerint lebontja a kapcsolatot.
- Kapcsolatmentes protokoll (*connectionless protocol*) használata esetén a küldő egyszerűen elküldi az adatokat, és maga a továbbiakban nem foglalkozik azzal, hogy ezeknek mi lett a sorsa, sőt, még csak nem is értesíti a fogadó felet arról, hogy adatokat szándékozik neki küldeni. Ez a kapcsolattípus tehát alapvetően egyirányú. A fogadó gép ehhez teljesen hasonlóan viselkedik. Ha érkezik hozzá adat, akkor fogadja, de erről a tényről semmiféle visszaigazolást nem küld az adatok forrásának.

A 6.1. ábrán két olyan embert látunk, akik kapcsolatközpontú kommunikációt folytatnak egymással. Ez a hasonlat természetesen nem arra hivatott, hogy ténylegesen szemléltesse a kapcsolatközpontú digitális kommunikáció minden mozzanatát, hiszen az ennél sokkal összetettebb, maga az alapkoncepció azonban ebből is világosan kiderül.

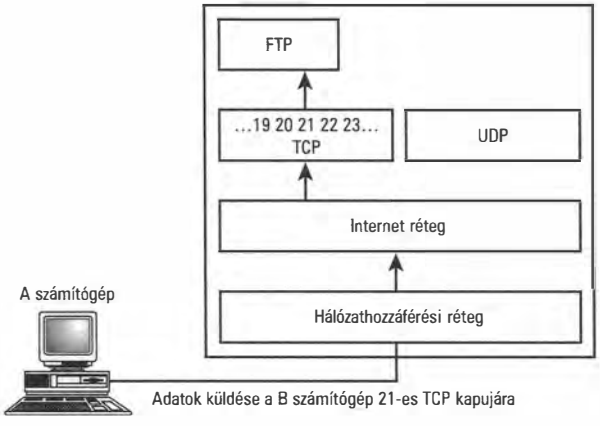
6.1. ábra
Kapcsolatközpontú kommunikáció

6.2. ábra
Kapcsolatmentes kommunikáció

6

Kapuk és foglalatok

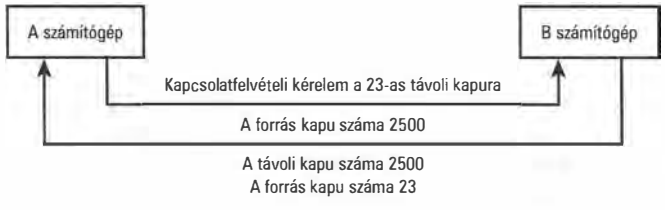
A szállítási réteg egy a hálózati alkalmazásokat és magát a hálózatot összekötő interfész, amely lehetővé teszi, hogy az átküldött adatokat a megfelelő alkalmazásokhoz továbbítsuk. A TCP/IP rendszerében az adatokat TCP és UDP protokollon is lehet továbbítani az egyik helyről a másikra, az alkalmazásokhoz továbbítás azonban mindkét esetben kapuk (port) segítségével történik. A kapu egy előre definiált belső cím, amelyen keresztül az alkalmazás elérheti a szállítási réteget, vagy fordítva (lásd a 6.3. ábrát). Az ügyfelek például egy távoli FTP kiszolgálót rendszerint a 21-es TCP kapun át érnek el.



6.3. ábra
Az adatokat a kapu száma (port address) alapján továbbítja a rendszer a megfelelő alkalmazáshoz.

Ha közelebbről megvizsgáljuk a szállítási réteg által alkalmazott címzési sémát, láthatjuk, hogy a TCP és UDP adatok továbbítása valójában nem is csak a kapukkal, hanem az úgynevezett foglalatokkal kapcsolatos, a címzés ténylegesen ezeken alapszik. A foglalat nem más, mint az IP cím és a kapuszám együttese. A 111.121.131.141.21-es foglalatszám például a 111.121.131.141 című számítógép 21-es kapuját jelenti.

A 6.4. ábra azt mutatja be, miként épül ki egy TCP kapcsolat két számítógép között a foglalatokra vonatkozó információ kicserélésével.



6.4. ábra
A forrás- és célkapu számának kicserélése két egymással kommunikáló számítógép között.

A következőkben megvizsgáljuk, pontosan hogyan is éri el egy számítógép egy másik számítógép egy adott hálózati alkalmazását a kapuszámok segítségével.

1. Az „A" számítógép kapcsolatot kezdeményez egy a „B" számítógépen futó hálózati alkalmazással egy úgynevezett „jól ismert" kapun (well-known port) keresztül. A jól ismert kapu nem más, mint egy szám, amit a IANA (Internet Assigned Numbers Authority) nevű szervezet egy bizonyos alkalmazástípushoz hozzárendelt. A leggyakoribb ilyen „jól ismert" TCP és UDP kapuszámokat a 6.1. ás a 6.2. Táblázatban foglaltuk össze. Összemásolva a „B" számítógép IP címét és a megfelelő jól ismert kapu számát az „A" gép megkapja a célgép foglalatszámát. Az „A" által küldött kapcsolatot kezdeményező adattömbbe ezen az információn

kívül bekerül annak a helyi kapunak a száma is, amelyet a „B" gépnek kell használnia, ha adatokat küld vissza. Ez az „A" gép forrásfoglalatának címe (source socket address).

2. A „B" számítógép veszi a jól ismert kapun keresztül az „A" gép kapcsolatfelvételi kérelmét, és a csomagban talált információ alapján visszaküldi a saját adatait az „A" gép forráscímére. A „B" gép szempontjából természetesen ez a foglalatcím lesz a célcím az „A" gépen.

A TCP kapcsolatok működéséről ebben az órában még részletesebben is lesz szó.

6.1. Táblázat *A legfontosabb „jól ismert" TCP kapuk*

Szolgáltatás	TCP kapuszám	Rövid leírás
tcpmux	1	TCP kapuszolgáltatás többszöröző (TCP port service multiplexer)
compressnet	2	Kezelő alkalmazás (Management utility)
compressnet	3	Tömörítő alkalmazások (Compression utility)
echo	7	Visszhang (Echo)
discard	9	Elvetés vagy null (Discard or null)
systat	11	Felhasználók (Users)
daytime	13	Idő szolgáltatás (Daytime)
netstat	15	Hálózati állapot (Network status)
qotd	17	A nap idézete (Quote of the day)
chargen	19	Karaktergenerátor (Character generator)
ftp-data	20	Fájlátvitel adatai (File Transfer Protocol data)
ftp	21	Fájlátvitel vezérlése (File Transfer Protocol control)
ssh	22	Biztonságos héj (Secure Shell)
telnet	23	Hálózati terminál szolgáltatás (Terminal network connection)
smtp	25	Egyszerű levéltovábbítási protokoll (Simple Mail Transport Protocol)
nsw-fe	27	NSW felhasználói rendszer (NSW user system)
time	37	Időszolgáltatás (Time server)
name	42	Gépnévszolgáltatás (Host name server)
domain	53	Tartományi névszolgáltatás (Domain name server ; DNS)
gopher	70	Gopher szolgáltatás (Gopher service)
finger	79	Finger (letapogatás) szolgáltatás (Finger)
http	80	Webszolgáltatás (WWW service)
link	87	Teletype kapcsolat (TTY link)
supdup	95	SUPDUP protokoll

6.1. Táblázat *A legfontosabb „jól ismert" TCP kapuk (folytatás)*

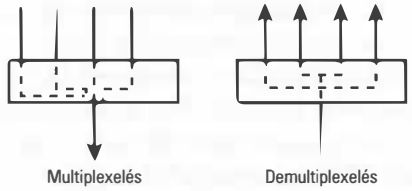
Szolgáltatás	TCP kapuszám	Rövid leírás
pop	109	POP protokoll (Post Office Protocol)
pop2	109	POP2 protokoll (Post Office Protocol 2)
pop3	110	POP3 protokoll (Post Office Protocol 3)
auth	113	Hitelesítési szolgáltatás (Authentication service)
sftp	115	Biztonságos fájlátvitel(Secure FTP)
uucp-path	117	Unix to Unix Copy Protokoll irányítási szolgáltatás (UUCP path service)
nntp	119	Usenet hírszolgáltatási protokoll (Usenet Network News Transfer Protocol)
nbsession	139	NetBIOS viszony szolgáltatás (NetBIOS session service)

6.2. Táblázat *A legfontosabb „jól ismert" UDP kapuk*

Szolgáltatás	TCP kapuszám	Rövid leírás
echo	7	Visszhang (Echo)
discard	9	Elvetés vagy null (Discard or null)
systat	11	Felhasználók (Users)
daytime	13	Idő szolgáltatás (Daytime)
qotd	17	A nap idézete (Quote of the day)
chargen	19	karaktergenerátor (Character generator)
time	37	Időszolgáltatás (Time server)
domain	53	Tartománynév szolgáltatás (Domain name server ; DNS)
nameserver	53	Tartománynév szolgáltatás (Domain name server ; DNS)
bootps	67	Bootstrap protokoll szolgáltatás és DHCP (Bootstrap protocol service/DHCP)
bootpc	68	Bootstrap protokoll ügyfél és DHCP (Bootstrap protocol client/DHCP)
tftp	69	Triviális fájlátviteli szolgáltatás (Trivial File Transfer Protocol)
ntp	123	Hálózati idő szolgáltatás (Network Time Protocol)
nbname	137	NetBIOS név (NetBIOS name)
snmp	161	SNMP protokoll (Simple Network Management Protocol)
snmp-trap	162	SNMP csapda (Simple Network Management Protocol trap)

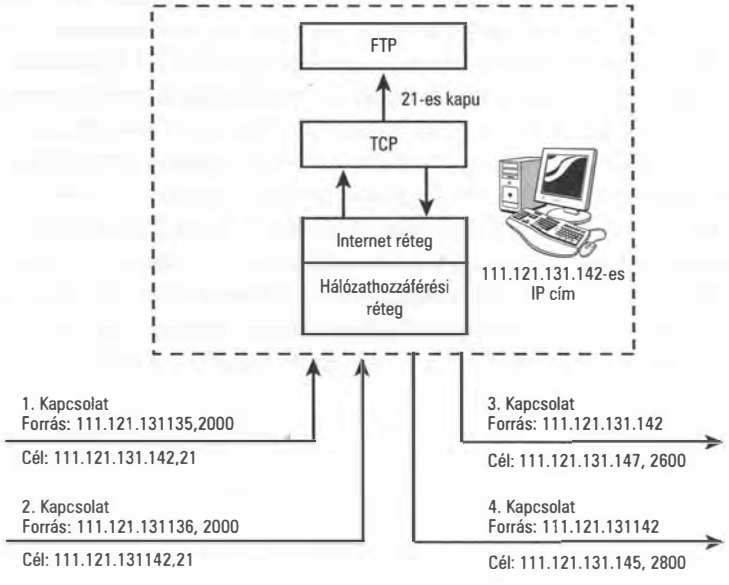
Multiplexelés a demultiplexelés

A fentiekben ismertetett foglalatcímzési mechanizmus mint a TCP, mind az UDP protokoll számára lehetővé teszi még egy fontos művelettípus végrehajtását: a szállítási réteg erre a mechanizmusra támaszkodva képes multiplexelési és demultiplexelési műveleteket végezni. Amint azt korábban már említettük, a multiplexelés (adategyesítés) olyan művelet, amelynél egy rendszer több forrásból kap adatokat, de ezeket egyetlen összetett (multiplexelt) kimenetként továbbítja. A demultiplexelés (adatfelbontás) ennek a folyamatnak az ellentéte, vagyis amikor a rendszer egyetlen bemenetről fogad adatokat, de ezeket szétbontva több helyre továbbítja (lásd a 6.5. ábrát).



6.5. ábra
A multiplexelés és demultiplexelés lényege

A multiplexelés és demultiplexelés képessége esetünkben azért fontos, mert ez teszi lehetővé, hogy a TCP/IP verem alsóbb rétegei egységesen tudják feldolgozni a kapott adatokat, függetlenül attól, hogy azok eredetileg milyen hálózati alkalmazástól származnak. Az eredeti alkalmazásokkal való kapcsolat számontartása egyedül a szállítási réteg feladata. Az internet réteg már egyetlen „vastag" csövön át kapja az információt, amely számára független a hálózati alkalmazásoktól.



6.6. ábra
A foglalatcím egyedileg azonosítja egy kiszolgáló egy adott szolgáltatását.

A multiplexelés és demultiplexelés kulcsa a foglalatcím. Mivel a foglalatcím egyaránt tartalmazza az IP címet és a kapuszámot, egyedileg képes azonosítani egy adott számítógépen futó adott alkalmazást. Nézzük meg például a 6.6. ábrán vázolt FTP kiszolgáló működését. A kliensgépek valamennyien a jól ismert 21-es kapuszámot használják a kommunikáció során, a cél-foglalatcímük (destination socket address) azonban különböző és garantáltan egyedi. Ehhez teljesen hasonlóan a kiszolgáló oldalán is igaz, hogy bár a rajta futó valamennyi alkalmazás a kiszolgáló IP címét használja a kommunikáció során, a 21-es TCP kapun át csak az FTP szolgáltatás érhető el, semmi más.

A TCP és UDP protokollok működése

Amint azt korábban már említettük, a TCP kapcsolatközpontú protokoll (*connection-oriented*), amely kiterjedt hibaellenőrzést és folyamatszabályozást tesz lehetővé. Az UDP ezzel szemben kapcsolatmentes protokoll (*connectionless*), amely sokkal kevésbé kifinomult szolgáltatásokat nyújt a hálózati hibák kezelése. Úgy is fogalmazhatnánk, hogy a fő tervezési cél a TCP esetében a biztonság, az UDP esetében pedig a sebesség volt. Az interaktív hálózati alkalmazások – ilyen például a Telnet vagy az FTP – általában a TCP protokollt használják. Az olyan alkalmazások viszont, amelyek a hibaellenőrzést maguk végzik, vagy a működésükhöz egyszerűen nincs szükség különösebb hibakezelésre, általában az UDP protokollra támaszkodnak.

Amikor egy fejlesztő egy hálózati alkalmazást tervez, szabadon eldönthet, hogy átviteli protokollként a TCP-t vagy az UDP-t kívánja használni. Az UDP-nek mint említettük sokkal egyszerűbb a vezérlése is, ez azonban egyáltalán nem föltétlenül jelenti azt, hogy az UDP rosszabb lenne, vagy valamiféle korlátot jelentene az alkalmazások minőségével kapcsolatban. A TCP által alapból biztosított ellenőrzések bizonyos esetekben teljesen fölöslegesek. Azoknál az alkalmazásoknál pedig, amelyek működéséhez tényleg szükség van a kiterjedt hibakezelésre illetve folyamatszabályozásra, a fejlesztő még mindig dönthet úgy, hogy ezeket a funkciókat magán az alkalmazáson belül valósítja meg. Ennek megvan az az óriási előnye, hogy ezeket a funkciókat így speciálisan az adott felhasználási célra lehet optimalizálni, miközben átviteli protokollként használható a sokkal gyorsabb UDP. Ez olyannyira igaz, hogy bár az alkalmazási réteghez tartozó RPC (*Remote Procedure Call*) szolgáltatás segítségével a legkülönfélébb kifinomult hálózati alkalmazások valósíthatók meg, a távoli eljáráshívást használó programok fejlesztő gyakran döntenek úgy, hogy a hibakezelést és a folyamatszabályozást inkább megírják maguk, csak hogy ne lassítsák a hálózati kapcsolatot a TCP használatával. Hiába kapnának készen egy kész de lassú megoldást, inkább a gyorsabb UDP-re támaszkodnak.

TCP: A kapcsolatközpontú átviteli protokoll

A TCP kapcsolatközpontú viselkedéséről az óra korábbi szakaszaiban tulajdonképpen már esett szó. It és most néhány egyéb, érdekes szolgáltatását említjük meg.

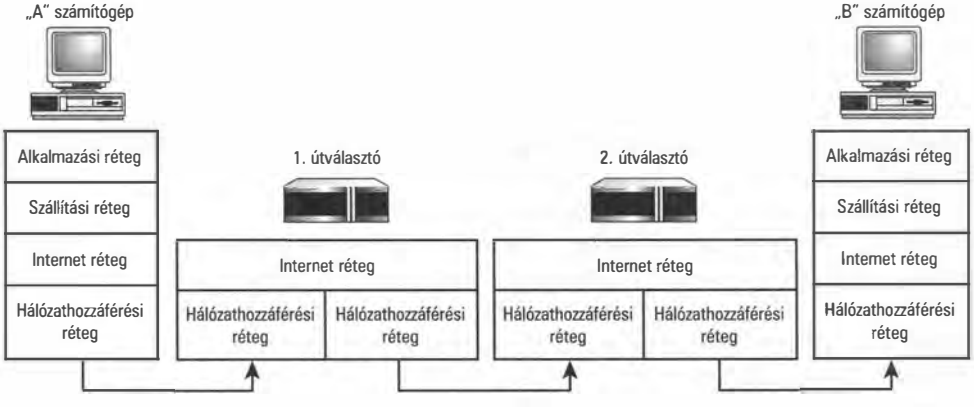
- **Adatfolyam-központú feldolgozás (*stream-oriented processing*)** – A TCP az adatokat folyamként kezeli. Ez azt jelenti, hogy a TCP protokoll akár bájtonként is képes adatokat fogadni, nem csak előre meghatározott méretű és formátumú csomagok formájában. A kapott adatokból aztán változó hosszúságú szegmenseket képez, amelyeket az internet rétegnek továbbít.
- **Átsorolás (*resequencing*)** – Ha az adatcsomagok nem a megfelelő sorrendben érkeznek meg a címzetthez, a TCP képes a szakaszokat átsorolni, és visszaállítani az eredeti sorrendet.
- **Folyamatszabályozás (*flow control*)** – A TCP folyamatszabályozása gondoskodik róla, hogy az átvitel sebessége semmiképpen se haladhassa meg a fogadó gép feldolgozási képességét, vagyis ne keletkezhessen átviteli „túlfutás" (*overrun*). Ennek különösen a heterogén hálózatokban van nagy jelentősége, ahol az egyes gépek processzorainak sebessége, illetve átmeneti adattároló kapacitása jelentősen eltérhet.
- **Elsőbbségi viszonyok (*precedence*) és biztonság** – A DoD (Department of Defense) eredeti specifikációjában a TCP-vel kapcsolatban szerepelnek olyan szolgáltatások is, amelyek segítségével prioritási szintek állíthatók be, illetve biztonságosabbá tehető az átvitel. Ugyanakkor ezeket a szolgáltatásokat számos TCP megvalósítás nem tartalmazza.
- **Kapcsolatok tiszta bontása (*graceful close*)** – A TCP a kapcsolatokat ugyanolyan gonddal bontja le, mint ahogy megnyitja őket. Ez elsősorban azt jelenti, hogy a kapcsolat bontása előtt gondoskodik az összes adatszegmens beérkezéséről.

Ha közelebbről megvizsgáljuk a TCP belső működését, azt láthatjuk, hogy a kommunikáció során két számítógép számos különféle kérelmet és visszaigazolást küld egymásnak a tényleges adatokon kívül. Ezek azok az extra üzenetek, amelyek segítségével a TCP megvalósítja a korábban említett kapcsolatközpontú kommunikációt a két fél között. A következő szakaszokban részletesebben is megvizsgáljuk a TCP által használt adatformátumot, az adatátvitel mikéntjét, illetve a TCP kapcsolatok felépítését és használatát. Előre kell bocsátanom, hogy ez meglehetősen sok műszaki részlet lesz, ám ez mindenképpen szükséges ahhoz, hogy láthassuk, egy protokoll nem egyszerűen egy adatformátum, hanem egymással kölcsönható folyamatok és eljárások egész rendszere, amely tervezésénél fogva egy jól meghatározott célt valósít meg.

Amint azt a 2. órában már tisztáztuk, egy réteges felépítésű protokollrendszer – mint amilyen a TCP/IP is – működésének lényege az, hogy a küldő és a fogadó gépen a megfelelő rétegek kommunikálnak egymással. Ez a gyakorlatban azt jelenti, hogy a küldő hálózathozzáférési rétege a fogadó hálózathozzáférési rétegével kommunikál, a küldő internet rétege a fogadó internet rétegével tartja a kapcsolatot, és így tovább.

Ennek megfelelően a küldő TCP szoftvere annak a gépnek a TCP szoftverével kommunikál, amelyikkel kapcsolatba szeretne lépni, vagy amellyel már élő kapcsolata van. Ha tehát egy a hálózati kommunikáció taglaló műszaki leírásban azt olvassuk, hogy az „A" számítógép TCP kapcsolatot épített ki a „B" számítógéppel, az valójában azt jelenti, hogy az „A" gép TCP szoftvere épített ki kapcsolatot a „B" gépe TCP szoftverével, amelyek ebben a kapcsolatban mindketten egy-egy helyi hálózati alkalmazás megbízottjaként működnek. Ennek a különbségtételnek igazából az első órában említett végponti ellenőrzés (*end-node verification*) szempontjából van különös jelentősége.

Emlékezzünk rá, hogy a TCP/IP játékszabályai szerint a kommunikáció ellenőrzéséről a végpontoknak kell gondoskodniuk. A „végpont" itt azt a két számítógépet jelenti, amelyek egymással kommunikálni kívánnak. Maguk az adatok köztes csomópontokon is áthaladhatnak, ezeknek azonban nincs „ellenőrzési kötelezettségük", ők csak továbbítják az adatokat. Egy tipikus kommunikációban az adatok egy alhálózatból egy másik alhálózat egy gépére kerülnek át néhány útválasztó érintésével (lásd a 6.7. ábrát). Ezek az útválasztók általában az internet rétegben működnek, vagyis egy szinttel az átviteli réteg alatt. Ez pedig azt jelenti, hogy egy tipikus útválasztónak fogalma sincs a szállítási réteg nyújtotta információkról. Semmi egyebet nem tesz, csak IP datagramokat formál a kapott adatokból és továbbadja azokat a megfelelő útvonalon. A TCP szegmensbe kódolt ellenőrzési és szabályozási információ kizárólag a fogadó gép TCP rétege számára jelent valamit. Ez a megoldás két szempontból is hasznos. Egyrészt az útválasztók gyorsabban dolgozhatnak, hiszen nem kell részt venniük a TCP kifinomult hibaellenőrzési és adatkezelési rituáléjában, másrészt így TCP/IP rendszeren belül megvalósul a DoD azon előírása, hogy ezekkel a feladatokkal kizárólag a végpontoknak kell foglalkozniuk.

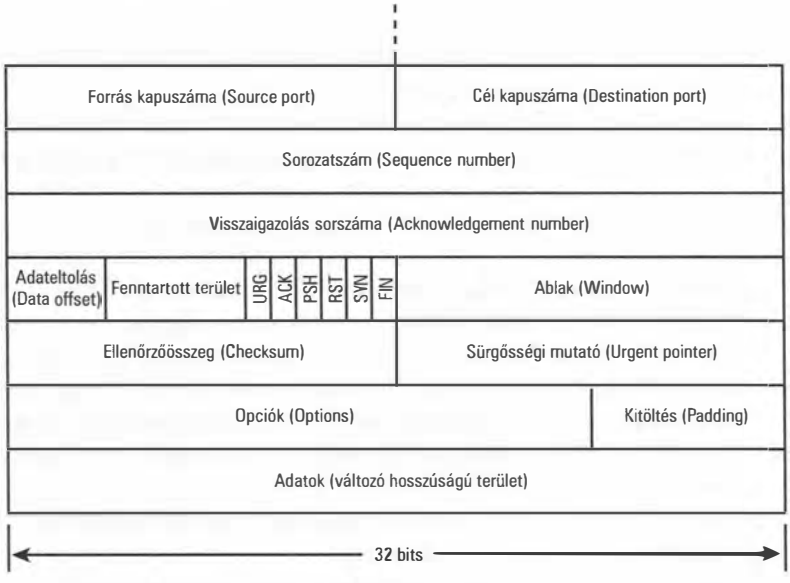


6.7. ábra

Az útválasztók továbbítják ugyan, de maguk nem dolgozzák fel a szállítási réteg adatait.

A TCP adatformátuma

A TCP adatformátumát vázlatosan a 6.8. ábra mutatja. Első ránézésre is látható, hogy itt egy viszonylag összetett adatszerkezetről van szó, ami nem véletlen, hiszen a TCP protokoll számos különféle funkciót lát el.



6.8. ábra
A TCP adatformátuma

A következőkben röviden áttekintjük az ábrán bemutatott valamennyi adatmező jelentését. Ezeknek az adatoknak a gyakorlati haszna amúgy sokkal világosabb lesz, miután elolvastuk a következő, a TCP kapcsolatokról szóló szakaszt is.

- **Forrás kapuszáma (*Source port*) (16 bit)** – A forrásgépen a kommunikáló hálózati alkalmazáshoz rendelt kapu száma.
- **Cél kapuszáma (*Destination port*) (16 bit)** – A célgépen a kommunikációban résztvevő alkalmazáshoz rendelt kapu száma.
- **Sorozatszám (*Sequence number*) (32 bit)** – Az adott szegmens első bájtjának sorozatszáma, kivéve, ha a SYN jelző be van állítva. Ha a SYN jelző értéke 1, akkor ennek a mezőnek a tartalma a kezdő sorozatszám (Initial Sequence Number; ISN), amit a rendszer arra használ, hogy szinkronizálja a sorozatszámokat a küldő és a fogadó gépen. Ilyen esetben az első oktet sorozatszáma eggyel nagyobb, mint a kezdő sorozatszám (vagyis ISN+1).
- **Visszaigazolás sorszáma (*Acknowledgement number*) (32 bit)** – A visszaigazolás száma tartalmazza annak a szegmensnek a sorozatszámát, amit a fogadó ebben az üzenetben visszaigazol. Az érték mindig eggyel nagyobb annak az utolsó bájtnak a sorozatszámánál, amit a fogadó visszaigazolt, vagyis valójában annak a következő bájtnak a sorozatszáma, amit a fogadó a küldőtől vár (utolsóként fogadott bájt sorozatszáma + 1).

- **Adateltolás (*Data offset*) (4 bit)** – Ez a mező mutatja meg a fogadónak, hogy milyen hosszú a fejléc, vagyis hol kezdődnek a hasznos adatok. Az adateltolást 32 bites szavakban kell érteni, vagyis ennek a mezőnek az értéke egy egész szám, ami úgy keletkezik, hogy a fejléc bitekben mért hosszát elosztjuk 32-vel.
- **Fenntartott terület (6 bit)** – Nevének megfelelően ez egy használaton kívüli terület, aminek az a rendeltetése, hogy a TCP protokoll jövőbeli fejlesztéseihez teret biztosítson. Valamennyi bitjének nullát kell tartalmaznia.
- **Szabályozó jelzők (*Control flags*) (mindegyik 1 bit)** – Ezek a jelzők speciális információkat közölnek a fogadóval az adott szegmenssel kapcsolatban.
  - **URG** – A jelző 1-es értéke jelzi, hogy a kérdéses adatszegmens kézbesítése sürgős, illetve hogy a sürgősségi mutató lényeges információt tartalmaz.
  - **ACK** – A jelző 1-es értéke jelzi, hogy a visszaigazolás sorszáma mező fontos információt tartalmaz.
  - **PSH** – Ha ennek a bitnek az értéke 1, az arra utasítja a TCP szoftvert, hogy minden adatot, amit eddig kapott, azonnal küldjön el a fogadónak (*push*).
  - **RST** – Ha értéke egy, a kommunikáló felek közti kapcsolat alaphelyzetbe áll (*reset*).
  - **SYN** – Ha ennek bitnek 1 az értéke, az egy kapcsolat kezdetét jelzi, vagyis azt, hogy a két félnek szinkronizálnia kell a sorozatszámait. Erről, illetve a háromutas kézfogásról (*three-way handshake*) a későbbiekben még részletesen lesz szó.
  - **FIN** – Ennek a bitnek az 1-es értéke azt jelzi, hogy a küldőnek nincs több továbbítandó adata, a kapcsolat lebontható.
- **Ablak (*Window*) (16 bit)** – Ezt a paramétert a rendszer a folyamatszabályozáshoz használja. Az ablakméret azoknak az utolsó visszaigazolt sorozatszámon túli sorozatszámoknak a tartományát határozza meg, amelyekhez tartozó adatokat a küldő továbbíthatja anélkül, hogy azokról visszaigazolást kapna.
- **Ellenőrzőösszeg (*Checksum*) (16 bit)** – Ennek a mezőnek a tatalmát a rendszer a továbbított adatok épségének ellenőrzésére használja. A fogadó gép ugyanazon algoritmus alapján, mint a küldő, maga is előállítja a kapott szegmensből az ellenőrzőösszeget, és összehasonlítja azt az ebben a mezőben található értékkel. Ha a kettő egyezi, az átvitel sikeres volt. Az ellenőrzőösszeg számításánál a TCP és az UDP is használ egy úgynevezett pszeudo fejlécet, amely IP címzési információt tartalmaz. Az UDP pszeudo fejlécről (pseudo-header) a későbbiekben még részletesen lesz szó.
- **Sürgősségi mutató (*Urgent pointer*) (16 bit)** – Ez egy eltolási mutató, amely arra a sorozatszámra mutat, ahonnan a sürgősen továbbítandó információ kezdődik.
- **Opciók (*Options*)** – A TCP-nek van néhány kisebb opcionális beállítási lehetősége. Ezek kerülnek ebbe a mezőbe.
- **Kitöltés (*Padding*)** – Amennyiben szükséges, ebbe a mezőbe a rendszer beilleszt néhány nulla bitet, hogy a hasznos adatok 32-bites szóhatáron kezdődjenek.
- **Adatok** – A szegmensben továbbítandó hasznos adatok.

A TCP-nek az összes itt felsorolt adatmezőre szüksége van ahhoz, hogy sikerrel tudja továbbítani, kezelni, ellenőrizni és visszaigazolni az adatokat. A következő szakaszban részletesebben is megvizsgáljuk, hogy a TCP szoftver működése, vagyis az adatok küldése és fogadása során mikor melyik mező tartalma kerül előtérbe, és pontosan mi a kérdéses adat jelentősége.

TCP kapcsolatok

Minden művelet, amit a TCP szoftver végrehajt, valamilyen kapcsolattal összefüggésben történik. A TCP kapcsolatokon keresztül küldi és fogadja az adatokat, ezeket a kapcsolatokat pedig a protokoll szabályai szerint kérelmezni kell, meg kell nyitni, a végén pedig szabályosan le kell zárni.

Amint arról korábban már volt szó, a TCP protokoll létének értelme az, hogy interfészt teremtsen az alkalmazások és a hálózat között, amelyen keresztül a programok elérik a hálózati elemeket. Ez az interfész a TCP kapukon (port) alapszik, ahhoz pedig, hogy a kapukon keresztül kommunikálni kezdhessünk, meg kell nyitni egy kapcsolatot az alkalmazás és a TCP között. A TCP alapvetően kétféle nyitott állapotot különböztet meg. Létezik passzív és aktív módon nyitott állapot.

- **Passzív módon nyitott állapot (*Passive open*)** – Ennél a módnál egy adott alkalmazás értesíti a TCP szoftvert, hogy a maga részéről készen áll bejövő kapcsolatok fogadására. A TCP ennek megfelelően kiépít vele egy kapcsolatot, és várja, hogy tényleg történjen valami, vagyis hogy valaki a hálózatról egy kérést küldjön a kérdéses alkalmazásnak.
- **Aktív módon nyitott állapot (*Active open*)** – Ilyenkor egy alkalmazás arra kéri a TCP szoftvert, hogy az építsen ki kapcsolatot egy passzív módon nyitott állapotban levő másik géppel. (Ami azt illeti, a TCP tulajdonképpen aktív készenléti állapotban levő géppel is képes kapcsolatot kiépíteni, feltéve, hogy a két fél egyszerre kezdeményezi a kapcsolatot.)

Általában ha egy alkalmazás kapcsolatokat kíván fogadni – ilyen lehet például egy FTP kiszolgáló – akkor magát és a hozzá rendelt TCP kaput passzív módon nyitott állapotba helyezi. Az ügyfél – esetünkben az FTP ügyfél – TCP állapota ezzel szemben mindaddig zárt lesz, amíg valamelyik felhasználó kapcsolatot nem kezdeményez a kiszolgálóval. Az ügyfélgép állapota ilyenkor átmegy aktív módon nyitottba. Az aktív módon nyitott állapotba kerülő számítógép (vagyis az ügyfél) ezután a megfelelő üzenetek cseréjével fölveszi a kapcsolatot a kiszolgáló TCP szoftverével. Ez a bizonyos kapcsolatkezdeményező információcsere az úgynevezett háromutas kézfogás (*three-way handshake*), amelyről a későbbiekben még részletesen lesz szó.

Ügyfélnek nevezzük azt a számítógépet, amely a hálózat egy másik gépét megszólítva attól szolgáltatásokat vár.

A kiszolgáló ezzel szemben olyan számítógép, amely a hálózat más gépeinek nyújt szolgáltatásokat.

A TCP szoftver változó hosszúságú adatszegmenseket képes küldeni. A szegmensen belül minden egy bájthoz tartozik egy sorozatszám (*sequence number*). A fogadó gépnek visszaigazolást kell küldeni minden egyes bájtról, amit sikeresen megkapott. A TCP kommunikáció tehát nem egyéb, mint átvitelek és visszaigazolások sorozata. Az átvitel előrehaladását a TCP szoftver a TCP fejlécekben található sorozatszám (*sequence number*) és visszaigazolási sorszám (*acknowledgement number*) alapján tudja követni (lásd a korábbi vázlatos felsorolást).

A fejléc valójában nem tartalmazza minden egy átvitt bájt sorozatszámát. Ehelyett a fejlécben található sorozatszám mező az adott szegmensben található adatok első bájtjának sorozatszámát tárolja.

Ez utóbbi szabály alól csak egy kivétel van, nevezetesen ha az adott szegmens egy kommunikáció első szegmense (részletesebben lásd a háromutas kézfogás leírásában). Ilyenkor a sorozatszám mező az úgynevezett kezdő sorozatszámot (*Initial Sequence Number; ISN*) tartalmazza, amely eggyel kisebb, mint az adatok első bájtjának sorozatszáma. (Vagyis az első bájt sorozatszáma ilyenkor ISN+1.)

Ha az adatszegmens sikeresen megérkezett a fogadóhoz, akkor a fogadó azt visszaigazolja. Ehhez a fejléc visszaigazolási sorozatszám (akcnowledgement number) mezőjét használja, amelyben az utolsó sikeresen fogadott bájt sorozatszámánál eggyel nagyobb számot helyez el. Másként fogalmazva ez a mező annak a bájtnak a sorozatszámát tartalmazza, amelynek fogadására a másik fél felkészült.

Ha a visszaigazolás adott időn belül nem érkezik meg a küldőhöz, akkor az újraküldi az adatokat úgy, hogy az új szegmensben az utoljára visszaigazolt bájt utáni bájttól kezdődően helyezi el az átvinni kívánt adatokat.

Kapcsolat felépítése

Ahhoz, hogy a fent vázolt sorozatszámokon illetve visszaigazolásokon alapuló rendszer működőképes legyen, a küldő és a fogadó számítógépnek szinkronizálnia kell az alkalmazott sorozatszámokat. Másként fogalmazva egy kommunikációban „B" számítógépnek tudnia kell, milyen kezdő sorozatszámot (Initial Sequence Number; ISN) használt „A" számítógép, mikor megnyitotta a kapcsolatot. Teljesen hasonlóan „A" gépnek is tudnia kell, hogy „B" gép honnan kezdte a számozást, máskülönben nem tudja értelmezni a visszafelé áramló adatokat.

A sorozatszámok szinkronizálást szokás háromutas kézfogásnak (*three-way handshake*) is nevezni. Ez a bizonyos háromszoros kézfogás minden TCP kommunikáció elején lezajlik, három lépése pedig a következő.

1. Az „A" számítógép elküld egy szegmenst „B" számítógépnek, amelyben:
   SYN = 1
   ACK = 0
   Sorozatszám (Sequence Number) = X (ahol X az „A" gép kezdő sorozatszáma)

   Az aktív módon nyitott állapotban levő számítógép (jelen esetben „A") elküld egy szegmenst a másik félnek, amelyben a SYN jelző értéke 1, az ACK jelző értéke pedig 0. A SYN az angol synchronize (szinkronizálás) rövidítése, vagyis azt jelzi, hogy az „A" gép kapcsolatot próbál kiépíteni. Ennek az első szegmensnek a fejléce tartalmazz a kezdő sorozatszámot (ISN) is, ami az első azon sorozatszámok közül, amelyeket az „A" gép a kommunikáció során használni fog. A „B" gépnek küldött első bájt sorozatszáma ISN+1 lesz.
2. A „B" számítógép fogadja „A" gép üzenetét, és visszaküld neki egy szegmenst, amelyben
   SYN = 1 (még mindig a szinkronizálás fázisánál tartunk)
   ACK = 1 (a visszaigazolási mező releváns információt tartalmaz)
   Sorozatszám = Y (ahol Y a „B" számítógép ISN-je)
   Visszaigazolási sorszám = M + 1 (ahol M az „A" géptől kapott utolsó sorozatszám)
3. Az „A" gép elküld „B" gépnek egy szegmenst, amelyben visszaigazolja „B" kezdő sorozatszámának fogadását:
   SYN = 0
   ACK = 1
   Sorozatszám = a sorban következő sorozatszám (jelen esetben M + 1)
   Visszaigazolási sorozatszám = N + 1 (ahol N a „B" számítógéptől kapott utolsó sorozatszám)

A háromszoros kézfogás után a kapcsolat immár nyitva áll, a rendszer pedig küldhet és fogadhat adatokat a fönt vázolt sorozatszámok és visszaigazolások rendszerét használva.

A TCP folyamatszabályozási funkciói

A TCP fejlécben található ablakméret (*window*) paraméter lehetőséget teremt a folyamatszabályozásra. Ennek a paraméternek tulajdonképpen az a rendeltetése, hogy megakadályozzon egy olyan szituációt, amelyben a küldő túl gyorsan túl sok adatot próbál forgalmazni, a csomagok pedig tulajdonképpen azért vesznek el, mert a fogadó ezt az adatmennyiséget teljesítményéből adódóan képtelen ennyi idő alatt feldolgozni. A TCP által alkalmazott folyamatszabályozási mechanizmus az úgynevezett csúszó

ablak (*sliding window*) módszere. A fogadó számítógép az ablak (window) mezőben megadhat egy számot (ezt a szakirodalomban szokták pufferméret néven is említeni), amely azon sorozatszámok tartománya (ablaka), ameddig a küldő további adatokat küldhet az utolsó visszaigazolt sorozatszámon túl. Ha a küldő eléri ezt a határt, akkor megáll, és csak a következő visszaigazolás után küldhet újabb adatokat.

A kapcsolat lezárása

Ha elérkezik a kapcsolat lezárásának ide, akkor az „A" számítógép elhelyez az adatfolyamban egy olyan szegmenst, amelyben a FIN jelző értéke 1-re van állítva. A kommunikációt végző alkalmazás ezután a lezáró várakozás állapotába (*fin-wait state*) kerül. Ebben az állapotban az „A" gép TCP szoftvere tovább folytatja a szegmensek küldését és fogadását, vagyis feldolgozza a már a pufferbe kerül adatokat, de új adatot már nem fogad az alkalmazástól. Amikor a „B" számítógép megkapja a FIN jelzőt tartalmazó csomagot, visszaigazolja azt, elküldi a még várakozó szegmenseket és értesíti a helyi alkalmazást, hogy FIN jelet kapott. Ezután maga is elküld egy FIN szegmenst az „A" gépnek, ezt az „A" szintén visszaigazolja és ezzel a kapcsolat lezárul.

UDP: A kapcsolatmentes átviteli protokoll

Az UDP a TCP-nél sokkal egyszerűbb protokoll így a korábbi szakaszokban felsorolt funkciók közül gyakorlatilag egyiket sem tudja. Ugyanakkor azért van néhány olyan tulajdonsága, amelyeket ebben az órában érdemes megemlíteni.

Először is bár az UDP-t a legtöbbször olyan protokollként írják le, mint amelyhez egyáltalán nem tartozik hibaellenőrzési szolgáltatás, valójában képes végrehajtani egyfajta kezdetleges ellenőrzést az átvitt adatokon. Helyesebb tehát úgy fogalmazni, hogy az UDP egy korlátozott hibaellenőrzési képességekkel felruházott protokoll. Az UDP datagramok tartalmaznak egy ellenőrzőösszeget, amit a fogadó fél felhasználhat arra, hogy megbizonyosodjon az adatok épségéről. (A feltételes mód itt lényeges, ez a funkció ugyanis a fogadó oldalán legtöbbször kikapcsolható, és a fejlesztők gyakorta élnek is ezzel a lehetőséggel, hogy felgyorsítsák a kommunikációt.) Az UDP datagram ezen kívül tartalmaz egy úgynevezett pszeudo fejlécet, amelyben benne van a célgép címe. Ez lehetőséget teremt a téves helyre irányított datagramok felismerésére. Ha egy fogadó gép UDP modulja olyan datagramot kap, amely nem létező, vagy definiálatlan kapura irányul, akkor egy ICMP üzenetben tudatja a küldővel, hogy a kérdéses kapu nem elérhető.

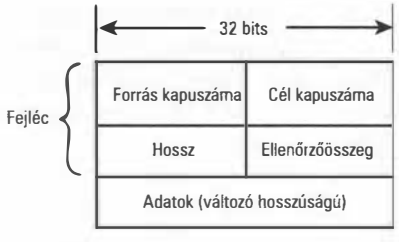
Másodszor az UDP nem rendelkezik azzal az átsorolási képességgel, amiről a TCP-vel kapcsolatban szó volt. A datagramok átrendezésének elsősorban az olyan nagy hálózatokban van jelentősége, mint amilyen például az internet, mivel az egyes adatcsomagok itt különböző utakon haladva juthatnak el a célgéphez, egyes útválasztókon komoly

késleltetést szenvedhetnek, így a sorrendjük összekeveredhet. Egy helyi hálózaton ezzel szemben ilyesmi nem nagyon történhet meg, így az UDP átsorolási képességének hiánya csak ritkán okoz problémát az adatátvitelben.

Az UDP-t könnyed, kapcsolatmentes kialakítása optimálissá teszi hálózati üzenetszórás (*broadcast*) megvalósítására. Üzenetszórásról akkor beszélünk, ha egy üzenetet az alhálózat valamennyi gépének meg kell kapnia, és fel kell dolgoznia. A korábban elmondottak alapján nyilván az olvasó számára is világos, hogy ha egy ilyen üzenetszóráshoz a küldőnek külön biztonságos TCP kapcsolatot kellene kiépítenie az összes címzettel, az igen komolyan csökkentené a hálózat működésének hatékonyságát.

Az UDP protokoll elsődleges célja, hogy a datagramokat az alkalmazási rétegnek eljuttassa. Maga az UDP meglehetősen keveset tesz, így fejlécének szerkezet is jóval egyszerűbb, mint a TCP esetében. Jellemző, hogy az UDP-t leíró RFC (RFC 768) mindössze három gépelt oldal. Amint azt korábban is említettük, az UDP nem küldi újra a sérült vagy hiányzó datagramokat, nem javítja meg a rossz sorrendben érkező adatok sorrendjét, nem dobja el a duplán beérkezett datagramokat, nem igazol vissza sikeresen vett adatot, nem épít fel kapcsolatot a kommunikáló felek között és nem is bontja le azt. Az UDP tehát egy olyan végtelenül egyszerű kommunikációs csatorna, amelyen keresztül az alkalmazások a TCP nyújtotta számos szolgáltatás mellőzésével egymással adatot cserélhetnek. Ugyanakkor maga az UDP-t használó hálózati alkalmazás az említett funkciók közül bármelyiket megvalósíthatja, ha működéséhez erre szüksége van.

Az UDP fejléc négy egyenként 16 bites mezőből áll. Szerkezetét vázlatosan a 6.9. ábra mutatja.



6.9. ábra
Az UDP datagram fejléce és a hasznos adatok elhelyezkedése.

A következőkben megvizsgáljuk az egyes mezők jelentését.

- Forrás kapuszáma (*source port*) – Ez a mező foglalja el az UDP fejléc első 16 bitjét. Rendszerint annak az alkalmazásnak az UDP kapuszámát tartalmazza, amely a kérdéses datagramot küldi. Ugyanezt a kapuszámot fogja használni a címzett is, ha adatokat szeretne visszaküldeni a feladónak. Mindezzel együtt ez a mező valójában opcionális, vagyis a protokoll nem írja elő, hogy a küldnek a csomagban el kell helyeznie a saját kapuszámát. Ha a küldő nem adja meg magáról ezt az információt, akkor 16 darab nullás bitet kell ebbe a mezőbe írnia.

Ha a datagramban nincs megadva a forrás kapuszáma, akkor a fogadó értelemszerűen nem tud választ küldeni, ez azonban nem föltétlen baj. Számos egyirányú kommunikációs forma létezik ugyanis, ahol nincs szükség semmiféle válaszra.

- Cél kapuszáma (*destination port*) – Ebben a 16 bites mezőben szerepel a címzett kapuszáma. Az UDP szoftvernek ide kell továbbítania a datagramot.
- Hossz – Ebben a 16 bites mezőben szerepel az UDP datagram oktetekben mért hossza. A hosszba beleszámít maga a fejléc is. Mivel ennek a hossza nyolc oktet, az ebben a mezőben megadott érték mindig nagyobb, mint nyolc.
- Ellenőrzőösszeg (*checksum*) – Ennek a 16 bites mezőnem az értékét használhatja a fogadó fél annak eldöntésére, hogy a datagram megsérült-e az átvitel során. Az ellenőrzőösszeg egy olyan speciális művelet eredménye, amit a rendszer bináris adatok sorozatán hajt végre. Az UDP esetében a számítás alapja a pszeudo fejléc, maga az UDP fejléc, az UDP datagramban található adatok, valamint esetleg egy térkitöltésre használt csupa nulla bitekből álló oktet. Ez utóbbira ahhoz van szükség, hogy a számítás bemenete mindenképpen páros számú oktetből álljon, mivel az algoritmus ezt megköveteli. Az ellenőrzőösszeget a küldő és a fogadó egyaránt kiszámítja. Ha a fogadó oldalán az eredmény nem egyezik meg a csomagban talált értékkel, akkor az adatok az átvitel során megsérültek.

Mivel az UDP datagramok fejléce sem a cél, sem a forrás IP címét nem tartalmazza, előfordulhat, hogy a csomag rossz helyre (rossz géphez vagy nem a megfelelő szolgáltatáshoz) kerül. Az ellenőrzőösszeg kiszámításához használt úgynevezett pszeudo fejléc (*pseudo-header*) ugyanakkor tartalmazza a címzett IP címét, így a fogadónak van esélye arra, hogy kiszűrje az adatfolyamból a valójában nem neki szóló datagramokat.

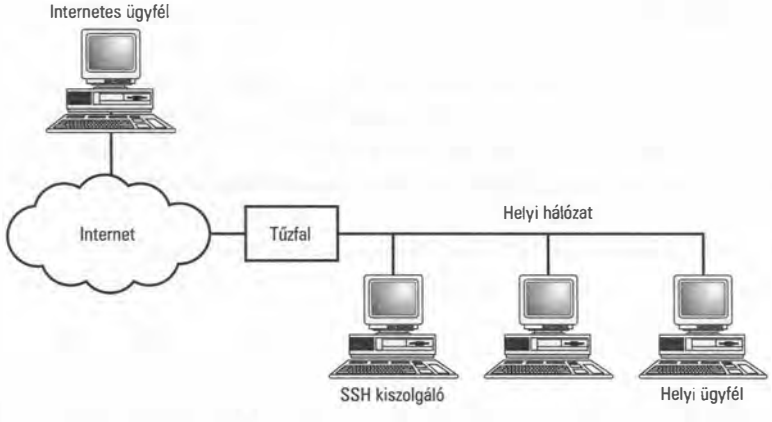
A szállítási réteghez a TCP-n és UDP-n kívül valójában tartozik még néhány ritkábban használt protokoll. Ilyen például a DCCP (Datagram Congestion Control Protocol), vagy az SCTP (Stream Control Transmission Protocol), amelyek olyan kifinomultabb adatkezelési és kommunikációs szolgáltatásokat nyújtanak, melyek a TCP és UDP repertoárjában nem szerepelnek. Említésre érdemes még az RTP (Realtime Transport Protocol), amelyet élő hang és kép közvetítésére lehet használni.

Tűzfalak és kapuk

A tűzfal (*firewall*) olyan eszköz, amely a helyi hálózatot védi az illetéktelen behatolásoktól. Maga az elnevezés afféle internetes szakzsargon, hiszen a szó szoros értelmében nincs semmi köze se a tűzhöz, se a falhoz, ráadásul ami a definícióját illeti, még az sem egységes. A tűzfalak ugyanis számos különféle funkciót láthatnak el. Ugyanakkor van egy olyan közös funkciójuk, amely igen szoros kapcsolatban áll a jelen óra anyagával, tehát érdemes már itt megemlíteni.

Ez a bizonyos közös funkció a tűzfalak azon képessége, hogy blokkolni tudják a külső hozzáférést megadott TCP és UDP kapukhoz. Ami azt illeti, az angol szakzsargonban a tűzfal szót néha még igeként is használják (*to firewall*). Ilyenkor egyszerűen azt jelenti, hogy kizárjuk/kiszűrjük az egy adott kapura irányuló hozzáférési kísérleteket.

Tegyük fel például, hogy egy ügyfélgép biztonságos héjat (SSH) szeretne nyitni egy kiszolgálón. Ehhez először is üzenetet kell küldenie az SSH „jól ismert" kapujára, ami nem más, mint a 22-es TCP kapu. (Az SSH-ról amúgy a 15. órában még részletesen is lesz szó.) Mármost ha egy rendszergazda fél attól,hogy illetéktelenek férhetnek hozzá az általa üzemeltetett kiszolgálóhoz SSH-n keresztül, akkor bezárhatja a 22-es TCP kaput, ami persze egyenértékű azzal, hogy nem használja az SSH szolgáltatást. Ez tökéletes megoldás abban az esetben, ha tényleg senkinek nincs szüksége erre a szolgáltatásra, de kétségkívül megvan az a hibája, hogy azokat is kizárja, akiknek amúgy joguk lenne hozzáférni a kérdéses géphez. (Mert ugyebár mire való egy kiszolgáló, ha nem használjuk?) Az igazi megoldás ilyenkor egy tűzfal beüzemelése, amely kiszűri a kiszolgáló 22-es TCP kapujára a külvilágból irányuló kéréseket (lásd a 6.10. ábrát). A helyi hálózat felhasználóinak forgalma nem halad át a tűzfalon, így ők továbbra is zavartalanul használhatják az SSH szolgáltatást, miközben a kiszolgáló is biztonságban van. A külvilág ezzel szemben nem érheti el a 22-es kaput, így nem kezdeményezhet SSH kapcsolatot a hálózat egyetlen gépével sem (tehát nem csak a kérdéses kiszolgálóval).



6.10. ábra
Tipikus példa tűzfal használatára.

A fenti leírásban említett 22-es TCP kapu természetesen csak egy példa. A gyakorlatban a tűzfalak az összes olyan kapura irányuló forgalmat szűrik, amelyeken keresztül potenciális támadás érkezhet. Mi több, sok hálógazda eleve úgy állítja be az általa használt tűzfalat, hogy az minden kaput szűrjön, kivéve azt a néhányat, amelyek használata elengedhetetlenül szükséges. Ilyen például a bejövő e-mail-ek kezeléshez szükséges kapu. Az is gyakori, hogy egy vállalat internetes jelenlétéért felelős eszközö-

ket – ilyen lehet például egy webszerver – a tűzfalon kívül helyezik el, hogy az ehhez való külső hozzáférések át se haladhassanak a belső hálózaton, s így semmiképpen ne jelenthessenek biztonsági kockázatot.

Ahogy egy tűzfal távol tarthatja a külső támadókat a helyi hálózattól, úgy megfelelő beállítások mellett a helyi hálózat felhasználóit is meggátolhatja abban, hogy bizonyos külső szolgáltatásokat használjanak.

Összefoglalás

Ebben az órában a TCP/IP szállítási rétegének bizonyos alapfunkcióit tekintettük át. Szó volt a kapcsolatközpontú és a kapcsolatmentes protokollokról, multiplexelésről és demultiplexelésről, kapukról és foglalatokról. Közelebbről is megismerkedtünk a TCP és UDP protokollal, megvizsgáltuk ezek néhány fontos szolgáltatását. Megtudtuk, hogy a TCP protokoll miként tesz eleget a végponti ellenőrzéssel kapcsolatos előírásoknak. Szó volt a TCP adatformátumáról, a folyamatszabályozásról, a hibaellenőrzésről, valamint a háromutas kézfogás folyamatáról, amit a TCP a kapcsolatok felépítése során használ. A TCP-n kívül megismerkedtünk az UDP fejléc felépítésével is.

Kérdések és válaszok

K *Miért van szükség a multiplexelésre és a demultiplexelésre?*

V Ha a TCP/IP nem nyújtana multiplexelési és demultiplexelési szolgáltatást, akkor egyszerre csak egy alkalmazás használhatná a hálózati szoftver szolgáltatásit, illetve egyszerre csak egy számítógép tarthatna kapcsolatot egy adott alkalmazással.

K *Miért használná egy szoftverfejlesztő az UDP protokollt a hálózati kommunikáció megvalósítására, ha egyszer a TCP sokkal jobb minőségű kapcsolatokat tesz lehetővé?*

V A TCP kifinomult minőségbiztosításának ára van: sokkal lassúbb rajta keresztül a kommunikáció. Ha tehát egy adott cél eléréséhez nincs föltétlen szükség a TCP nyújtotta hibaellenőrzésre és folyamatszabályozásra, akkor az UDP a jobb választás, mivel gyorsabb.

K *Miért van az, hogy az interaktív alkalmazások, mint például a Telnet vagy az FTP inkább a TCP protokollt használják, nem az UDP-t?*

V Az interaktív munkamenetekhez megbízható kapcsolatra van szükség, ezt pedig csak a TCP vezérlési és hibajavító szolgáltatásai képesek biztosítani.

K *Mi értelme van annak, hogy egy hálózati adminisztrátor egy tűzfal közbeiktatásával szándékosan meggátolja a hozzáférést egyes TCP vagy UDP kapukhoz?*

V A tűzfalnak az a feladata, hogy bizonyos kapuk blokkolásával meggátolja az internet felhasználóit abban, hogy a helyi hálózat gépeinek bizonyos szolgáltatásihoz hozzáférhessenek. A dolog fordítva is működhet, vagyis egy tűzfal a helyi hálózat felhasználóit is meggátolhatja abban, hogy bizonyos külső internetes szolgáltatásokat használjanak.

K *Miért nem küldenek az útválasztók TCP visszaigazolást azoknak a gépeknek, amelyek kapcsolatot akarnak létesíteni egy másik géppel?*

V Az útválasztók az internet rétegben működnek, vagyis fel sem dolgozzák a TCP szegmensekben kódolt információkat.

K *Egy működőképes FTP kiszolgáló jellemzően aktív módon nyitott, passzívan nyitott, vagy zárt állapotban van?*

V Egy működő FTP kiszolgáló normális esetben passzív módon nyitott állapotban várja a kapcsolatfelvételi kérelmeket.

K *Miért van szükség a háromutas kézfogás harmadik lépésére?*

V Az első két lépés megtétele után a két számítógép már kicserélte a kezdő sorozatszámokat (ISN) vagyis elvileg kezükben az összes lényeges információ, és készen állnak az adatcserére. Ugyanakkor az a számítógép, amelyik a második lépésben elküldte a kezdő sorozatszámát a partnerének, még nem kapott erről visszaigazolást, vagyis nem lehet benne biztos, hogy az információ célba ért. Éppen erre való a harmadik lépés, vagyis hogy a második lépésben küldött adatot a fogadó fél visszaigazolhassa.

K *Melyik az a mezője az UDP fejlécnek, amelynek kitöltése nem kötelező és miért?*

V Az egyetlen opcionális mező a forrás kapuszáma. Mivel az UDP kapcsolatmentes protokoll, a fogadó félnek nem kell ismernie a küldő kapuszámát. Ezt az információt csak akkor szokták megadni, ha a fogadó gépen futó alkalmazásnak kifejezetten szüksége van rá a hibaellenőrzés végett, vagy vagy valamilyen más célból.

K *Mi történik akkor, ha a forrás kapuszámának helyén 16 darab nullás bit szerepel?*

V Ez azt jelenti, hogy a fogadó gép nem fog tudni válaszolni a küldőnek, hiszen nem tudja, hova kellene küldenie azt.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **ACK** – Szabályozó jelző, amelynek 1-es értéke azt jelzi, hogy a TCP fejléc visszaigazolási sorozatszám (*Acknowledgement Number*) mezője lényeges információt tartalmaz.
- **Visszaigazolás sorozatszámát tartalmazó mező (*Acknowledgement Number field*)** – A TCP fejléc azon mezője, mely a fogadó számítógép által legközelebb várt bájt sorszámát tartalmazza. A visszaigazolási sorozatszám megadásával a címzett gyakorlatilag visszaigazolja a küldőnek a kérdéses sorozatszám előtti adatok fogadását.

- Aktív módon nyitott állapot (*Active open*) – Olyan állapot, amelyben a TCP szoftver kapcsolatot próbál kiépíteni egy távoli géppel.
- Ellenőrzőösszeg (*Checksum*) – Egy 16 bites számított érték, amit a fejléc is tartalmaz. Célja az adatátviteli hibák detektálása.
- Kapcsolatközpontú protokoll (Connection-oriented protocol) – Olyan protokoll, amely a kommunikáló felek között logikai kapcsolatot létesít, így vezérli a kommunikáció folyamatát.
- Kapcsolatmentes protokoll (*Connectionless protocol*) – Olyan protokoll, amely anélkül visz át a hálózaton adatokat, hogy a kommunikáló felek között logikai kapcsolatot létesítene.
- Szabályozó jelző (*Control flag*) – Olyan 1 bites jelző, amelynek értéke speciális információt hordoz a TCP szegmensről.
- Demultiplexelés (*Demultiplexing*) – Egyetlen bemenet szétválogatás utáni kiküldése több kimenetre.
- Cél kapuszáma (*Destination port*) – A fogadó gép azon TCP vagy UDP kapujának a száma, amelyre a kérdéses TCP szegmens vagy UDB datagram érkezni fog.
- FIN – Szabályozó jelző, melynek egyes értéke a TCP kapcsolat lebontását jelzi.
- Tűzfal (*Firewall*) – Olyan eszköz, amely a helyi hálózat gépeit védi az internetes felhasználók jogosulatlan hozzáférésétől.
- Kezdő sorozatszám (*Initial Sequence Number; ISN*) – Olyan szám, amely annak a számsorozatnak a kezdetét jelzi, amit egy TCP protokollon kommunikáló számítógép az átvitt adatok sorszámozására használ.
- Multiplexelés (*Multiplexing*) – Több bemenetből egyetlen kimenet előállítása.
- Passzív módon nyitott állapot (*Passive open*) – Olyan állapot, amelyben egy TCP kapu (általában egy kiszolgáló valamely szolgáltatása) készen áll a bejövő kapcsolatok fogadására.
- Kapu (*Port*) – Olyan belső cím, amely egy alkalmazást a helyi gép szállítási rétegével kapcsolja össze.
- Pszeudo fejléc (*Pseudo-header*) – Olyan az IP fejlécben található információkból levezetett adatszerkezet, amit a TCP és UDP ellenőrzőösszegek kiszámítása során használ a rendszer, illetve annak ellenőrzésére, hogy egy datagram nem került-e rossz helyre az IP fejlécben található információk téves módosulása miatt.
- Átsorolás (*Resequencing*) – Az a folyamat, amelyben a rendszer visszaállítja a rossz sorrendben beérkezett TCP szegmensek eredeti sorrendjét.
- Szegmens (*Segment*) – Egy TCP fejléc és a hozzá tartozó hasznos adatok együttese.
- Sorozatszám (*Sequence number*) – Egyedi sorozatszám, amely egy TCP protokollon átküldött bájtot azonosít.
- Csúszó ablak (*Sliding window*) – A sorozatszámok egy olyan tartománya, amelyekhez tartozó adatok átküldését a fogadó fél engedélyezte a küldőnek. A TCP protokoll a csúszó ablak módszert használja folyamatszabályozásra.
- Foglalat (*Socket*) – Egy adott gépen futó adott alkalmazást egyértelműen azonosító hálózati cím, amely a számítógép IP címéből és az alkalmazás kapuszámából áll.
- Forrás kapuszáma (*Source port*) – Annak a számítógépnek a kapuszáma, amely a kérdéses TCP szegmenst vagy UDP datagramot küldi.

- **Adatfolyam-központú bemenet (*Stream-oriented input*)** – Folyamatos (bájtonkénti) bemenet. Ellentéte az előre megadott méretű blokkokban történő kommunikációnak.
- **SYN** – Olyan szabályozó jelző, amelynek 1-es értéke jelzi, hogy a sorozatszámok szinkronizálása folyamatban van. A SYN jelzőt a TCP kapcsolatok felépítése során használja a rendszer a háromutas kézfogás részeként.
- **TCP** – A TCP/IP protokollcsomag megbízható, kapcsolatközpontú adatátviteli protokollja.
- **Háromutas kézfogás (*Three-way handshake*)** – Egy három lépésből álló eljárás, amely során az egymással kommunikálni kívánó számítógépek szinkronizálják a sorozatszámaikat, és felépül köztük egy TCP kapcsolat.
- **UDP** – A TCP/IP protokollcsomag nem megbízható, kapcsolatmentes adatátviteli protokollja.
- **„Jól ismert" kapuk (*Well-known ports*)** – A leggyakoribb hálózati alkalmazások előre meghatározott kapuszámai. A jól ismert kapuszámokat a IANA (Internet Assigned Numbers Authority) határozza meg.

7. ÓRA

Az alkalmazási réteg

Ebben az órában a következőkről lesz szó:

- Hálózati szolgáltatások
- Programozási felületek (API-k)
- TCP/IP segédprogramok (*utilities*)

A TCP/IP verem tetején található az alkalmazási réteg, amely a szállítási réteg fölé épített különböző hálózati alkotóelemek laza gyűjteménye. A következő órán ezen alkotóelemek közül veszünk szemügyre néhányat. Szót ejtünk arról is, hogy ezek az alkotóelemek hogyan viszik közelebb a hálózathoz a felhasználót. Megvizsgáljuk az alkalmazási réteg szolgáltatásait, működési környezeteit és hálózati alkalmazásait.

Az óra anyagának elsajátítása után az olvasó képes lesz

- jellemezni az alkalmazási réteget
- jellemezni az alkalmazási réteg néhány hálózati szolgáltatását
- felsorolni a TCP/IP néhány fontos segédprogramját

Mi is az az alkalmazási réteg?

Az alkalmazási réteg a TCP/IP protokollkészlet legfelső rétege. Az alkalmazási rétegben található néhány olyan hálózati alkalmazás és szolgáltatás, amely TCP és UDP kapukon keresztül kommunikál az alacsonyabban fekvő rétegekkel (ahogy arról a 6. órán, „A szállítási réteg" c. fejezetben szó volt). Felmerülhet a kérdés, hogy miért is tekintjük egyáltalán a veremhez tartozónak az alkalmazási réteget, hiszen más rétegek (például a TCP és az UDP kapuk) jól definiált hálózati felületet jelentenek. Fontos felidéznünk azonban azt is, hogy egy többrétegű architektúra esetén (mint amilyen a TCP/IP) *minden* réteg felületet biztosít a hálózathoz. Az alkalmazási rétegnek ugyanúgy ismernie kell a TCP és UDP kapukat, mint ahogy a szállítási réteg is ismeri őket, és a különféle adatokat alkalmas módon kell csatornákba terelnie.

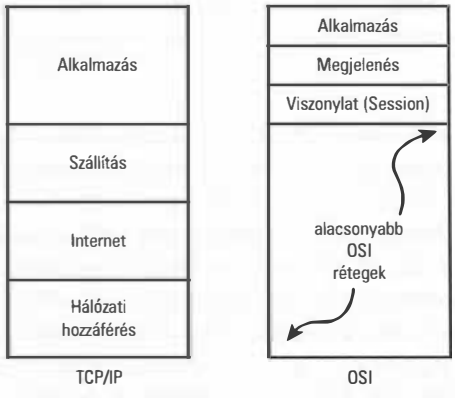
A TCP/IP alkalmazási rétege nem más, mint hálózati használatra készített szoftverelemek halmaza, amelyek a TCP és UDP kapuknak (és kapuktól) küldenek (és kapnak) információkat. Az alkalmazási rétegnek ezek az elemei nincsenek egy szinten annyiban, hogy logikailag (vagy szerepüket illetően) nem mindig állíthatóak párhuzamba egymással. Néhány alkotóelem egyszerű segédprogram, amely csak információt gyűjt a hálózati beállításokról. Más programok viszont felhasználói felületként működnek (ilyen például az X Window rendszer). Vannak alkalmazásprogramozási felületek (API-k) is, amelyek asztali operációs rendszer környezetek támogatására készültek. Vannak a rétegnek olyan alkotóelemei, amelyek hálózati szolgáltatásokat biztosítanak, mint amilyen például a nyomtatás vagy a névfeloldás. (A névfeloldásról részletesen tanulunk majd a 11. fejezetben („A névfeloldás".) Ezen az órán betekintést nyerünk néhány olyan szolgáltatásba és alkalmazásba, amelyek általában megtalálhatóak az alkalmazási réteg elemei között. Ezeknek az elemeknek a konkrét megvalósítása természetesen erősen függ a használt szoftverkörnyezettől.

Ezek után lássunk egy gyors összehasonlítást: vessük össze a TCP/IP alkalmazási rétegét az OSI modell neki megfelelő rétegével.

A TCP/IP alkalmazási réteg és az OSI

Ahogy azt a 2. órán („Hogyan működik a TCP/IP") említettük, a TCP/IP nem felel meg pontosan a hétrétegű OSI hálózati modellnek. Az OSI modell azonban kihatással volt a különféle hálózati rendszerek kifejlesztésére. A jelenlegi tendenciák, amelyek a többprotokollos hálózati megoldások felé közelítenek, egyre inkább támaszkodnak az OSI kifejezésmódjára és elveire. Az alkalmazási réteg számos operációs rendszerben és hálózati környezetben működhet. Szinte mindegyikben hatékony eszköz lehet az OSI modell, ha a hálózati rendszerek definíciójáról vagy jellemzéséről van szó. Ha egy pillantást vetünk az OSI modell ábrájára, akkor könnyebben megérhetjük, hogy milyen folyamatok zajlanak a TCP/IP alkalmazási rétegében.

A TCP/IP alkalmazási réteg az OSI modellben az „alkalmazási", „megjelenési" és „viszony" rétegek egybefoglalásának felel meg (lásd a 7.1 ábrát). Az OSI modellnek ezek a finom rétegei (vagyis az egyetlen réteg helyett három réteg) további szerveződési lehetőségeket rejtenek; ezeket a TCP/IP elméleti szakemberei hagyományosan az „alkalmazásszintű (vagy folyamat-/alkalmazásszintű) szolgáltatások" fogalmával fejeztek ki.



7.1 ábra

Az alkalmazási réteg az OSI modell „alkalmazási", „megjelenési" és „viszony" rétegeinek felel meg.

Foglaljuk össze a TCP/IP alkalmazási rétegének megfelelő OSI rétegek szerepét:

- **Alkalmazási réteg** – Az OSI alkalmazási rétege (nem tévesztendő össze a TCP/IP alkalmazási rétegével) olyan alkotóelemekből áll, amelyek szolgáltatásokat nyújtanak a felhasználói alkalmazások számára és lehetővé teszik a hálózat elérését.
- **Megjelenési réteg** – A megjelenési réteg platformfüggetlen formátumúra alakítja az adatokat; ez végzi a titkosítást és az adattömörítést is.
- **Viszony réteg** – A viszony (*session*) réteg intézi a különféle alkalmazások közötti kommunikációt a hálózatra kötött számítógépeken. Ez a réteg olyan funkcionalitásokat is elérhetővé tesz a hálózati kapcsolatok számára, amelyek pusztán a szállítási réteg használatával nem lennének elérhetőek. Ilyen például a névfelismerés és a biztonság.

Ezek közül természetesen nem mindegyik szolgáltatást használja ki az összes alkalmazás és program. A TCP/IP modell szerint megírt programoknak nem feltétlenül kell igazodnia az OSI rétegeihez. Az OSI „alkalmazási", „megjelenési" és „viszony" rétegeinek feladatköreit viszont el kell látnia a TCP/IP alkalmazási rétegének.

Hálózati szolgáltatások

Az alkalmazási réteg összetevőinek jó része nem más, mint hálózati szolgáltatás. A korábbi órák során már említettük, hogy egy protokollrendszer bármely rétege szolgáltatásokat biztosít a rendszer más rétegei számára. Az esetek többségében ezek a szolgáltatások a protokollrendszer jól definiált, integráns részei. Az alkalmazási réteg esetében nem minden szolgáltatás szükséges a protokollszoftverek működéséhez. Inkább arra szolgálnak, hogy közvetlenül segítsék a felhasználót, vagy hogy a hálózathoz kapcsolják a helyi operációs rendszert.

El kell mondanunk, hogy a protokollverem alsóbb rétegei inkább a kommunikációs folyamatok technikai részleteihez kapcsolódnak, és nem jelentenek sokat egy hétköznapi felhasználó számára. Ezzel szemben az alkalmazási réteg számos olyan hálózati szolgáltatást magába foglal, amelyeket a felhasználók többnyire jól ismernek. Ilyen például a fájlkiszolgálás, a távoli elérési szolgáltatás, az email és a HTTP webes protokoll. Valójában könyvünk jelentős része az alkalmazási réteghez tartozó hálózati szolgáltatások részletezéséről szól. A 7.1 táblázatban megtaláljuk az alkalmazási réteg legfontosabb protokolljait és szolgáltatásait. Ezekről bőségesen fogunk tanulni a következő órákon. Most azonban tekintsük át az alkalmazási réteg legjelentősebb működési területeit, azaz:

- a fájlkiszolgáló és nyomtatási szolgáltatásokat
- a névfeloldási szolgáltatást
- a távoli elérést
- a webszolgáltatásokat

Az alkalmazási rétegben találunk más fontos hálózati szolgáltatásokat is (mint például a levelezőkiszolgálást és a hálózatkezelést), de ezeket egy másik alkalommal tárgyaljuk.

7.1 táblázat *Az alkalmazási réteg néhány protokollja*

Protokoll	Leírása
BitTorrent	P2P alapú (azaz közvetlenül egymással kommunikáló hálózati elemekből álló) fájlcserélő protokoll, amelyet gyakran használnak nagy fájlok internetről való letöltésére.
Common Internet File System (CIFS)	Az SMB fájlkiszolgáló protokoll továbbfejlesztett változata.
Domain Name System (DNS)	Több ezer kiszolgálóra elosztott hierarchikus adatbázisrendszer az internetes tartománynevek és IP-címek egymáshoz rendelésére.

7.1 táblázat *Az alkalmazási réteg néhány protokollja* *(folytatás)*

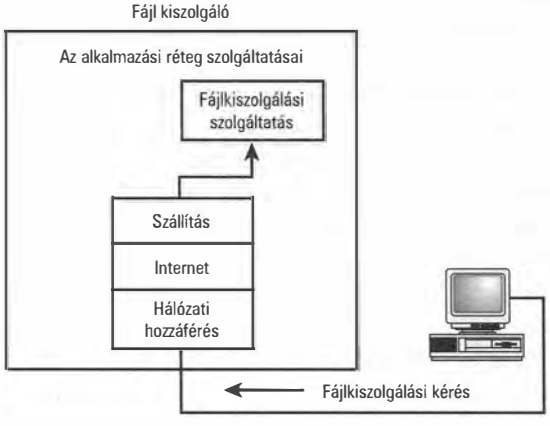
Protokoll	Leírása
Dynamic Host Configuration Protocol (DHCP)	Kliens-szerver protokoll, amely lehetővé teszi különféle hálózati paraméterek (például IP-cím) automatikus hozzárendelését az egyes hálózati gépekhez.
File Transfer Protocol (FTP)	Állományok fel- és letöltésére széles körben használt protokoll.
Finger	Felhasználói információk kinyerését lehetővé tevő protokoll.
Hypertext Transfer Protocol (HTTP)	A világháló (World Wide Web) kommunikációs protokollja.
Internet Message Access Protocol (IMAP)	Email üzenetek elérését lehetővé tevő közös protokoll.
Lightweight Directory Access Protocol (LDAP)	Címtárszolgáltatások létrehozását és használatát lehetővé tevő protokoll.
Network File System (NFS)	A hitelesített felhasználók számára távoli fájlelérést lehetővé tevő protokoll.
Network Time Protocol (NTP)	Órák (és más, idővel kapcsolatos erőforrások) TCP/IP hálózaton történő szinkronizálására szolgáló protokoll.
Post Office Protocol (POP)	Levelezőszerverről történő email-letöltésre használt protokoll.
Remote Procedure Call (RCP)	Távoli eljáráshívási protokoll, amely lehetővé teszi egy program számára egy másik számítógépen futó eljárás vagy program hívását.
Server Message Block (SMB)	Fájlkiszolgáló és nyomtatás-szolgáltatási protokoll.
Simple Network Management Protocol (SNMP)	Hálózati eszközök kezelését lehetővé tevő protokoll.

Fájlkiszolgáló és nyomtatási szolgáltatások

Kiszolgálónak (szerver) hívjuk az olyan számítógépet, amely más számítógépek számára szolgáltatást biztosít. A szolgáltatások két gyakori formája a fájlkiszolgáló és a nyomtatási szolgáltatás.

A nyomtatókiszolgáló (*print server*) nyomtató(ka)t kezel; az ide (különféle dokumentumok kinyomtatására) érkező kéréseket teljesíti. A fájlkiszolgáló adattároló eszközöket (például merevlemezeket) kezel, és ezen eszközökre érkező (írási és olvasási) kéréseket teljesít.

Mivel a fájlkiszolgáló és a nyomtatási szolgáltatások a leggyakoribb hálózati műveletek, gyakran együtt kezelik őket. Gyakran ugyanaz a számítógép (vagy akár ugyanaz a szolgáltatás) biztosítja a fájlkiszolgálást és a nyomtatási szolgáltatásokat is. Függetlenül attól, hogy együtt szervezzük-e őket, ugyanazon az elven működnek. A 7.2 ábra bemutat egy tipikus fájlkiszolgáló esetet. Az állománykérési információ hálózaton érkezik, és a különböző protokollrétegeken át eljut a szállítási réteghez, majd a megfelelő kapun keresztül átirányítódik (*routed*) a fájlkiszolgálóhoz.



7.2 ábra
Fájlkiszolgálás

A 7.2 ábra egy sematikus vázat mutat be az alapelemek kapcsolódásáról (és a TCP/IP-hez való kapcsolatukról). Egy valódi operációs rendszer valódi protokolljában még más rétegek, egyéb alkotóelemek is szerepelhetnek, amelyek részt vesznek az adatok eljuttatásában a fájlkiszolgáló szolgáltatáshoz.

A (különféle Unix/Linux operációs rendszereken alkalmazott) „Network File System" (NFS) és a (többnyire Microsoft operációs rendszerein futó) „Server Message Block" (SMB) fájlkiszolgáló rendszerek is az alkalmazási réteghez tartoznak, ahogy a klasszikus fájltovábbító segédprogramok, a „File Transfer Protocol" (FTP) és a „Trivial File Transfer Protocol" (TFTP) is.

Névfeloldási szolgáltatások

Ahogy azt az 1. órán már megtanultuk, a **névfeloldás** az a folyamat, amelynek során az IP-címekhez előre megadott, felhasználóbarát neveket rendelünk (melyek betűket és számokat és még néhány egyéb karaktert tartalmazhatnak). A tartománynév szolgáltatás (*Domain Name Service, DNS*) biztosítja az internet számára a névfeloldást; ám kisebb

(akár elszigetelt) TCP/IP hálózatokban is használható erre a célra. A DNS **névkiszolgálókat** (*name servers*) használ a DNS névlekérezések megválaszolására. A névkiszolgálás a névkiszolgáló számítógép alkalmazási rétegében fut; a program más névkiszolgálókkal is kommunikál a névfeloldási információkról. Léteznek másfajta névfeloldó rendszerek is, mint például a Network Information Service (NIS), a NetBIOS névfeloldás, valamint a Light Directory Access Protocol-lal (LDAP) kapcsolatos számos névfeloldási szolgáltatás-változat.

A névfeloldás kitűnő példája azoknak a szolgáltatásoknak, amelyek az alkalmazási rétegben futnak, és szervesen együttműködnek az alacsonyabb szintű protokollrétegekkel; ez a szolgáltatás is aktívan részt vesz a protokollverem kommunikációs folyamataiban. A DNS (vagy akár a WINS) lekérdezéseket az ügyfélgép protokollszoftvere kezdeményezi, nem pedig a felhasználó (vagy egy felhasználói alkalmazás). A felhasználó hivatkozik egy tartománynévre, és a mélyebben fekvő protokollszoftver (a névfeloldás használatával) átfordítja ezt a nevet a megfelelő IP-címre.

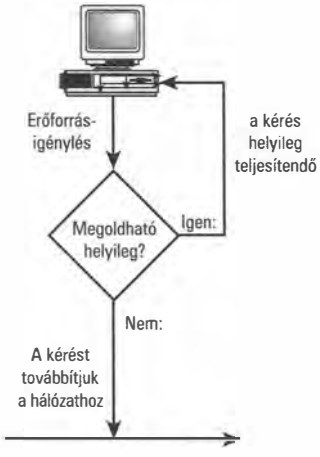
Távoli elérés

Az alkalmazási réteg számos olyan technológia működését biztosítja, amelyek lehetővé teszik a felhasználók számára, hogy egy számítógépről interaktív kapcsolatot kezdeményezzenek egy másik számítógéppel. Ahogy a 15. órán (a „Megfigyelés és távoli elérés" c. fejezetben) majd találkozni fogunk vele, a Telnet és az SSH (és még néhány hasonló eszköz) lehetővé teszi a felhasználó számára, hogy távolról bejelentkezzen egy rendszerbe, és ott a hálózaton keresztül parancsokat futtasson. Vannak modern képernyőmegosztó eszközök, amelyek hasonló bejelentkezési lehetőségeket biztosítanak az asztali felhasználói felületekhez is.

Néhány operációs rendszernek létezik egy **átirányító (redirector)** nevű szolgáltatása is, amely arra szolgál, hogy a helyi környezetet összehangolja a hálózattal. (Más néven ezt *requester*-nek, „igénylőnek" is szokták hívni.)

Ha az átirányító szolgáltatás-igénylést kap a helyi számítógéptől, akkor megvizsgálja, hogy a kérést ki tudja-e szolgálni a helyi gép erőforrásait igénybe véve, vagy a hálózat más gépéhez kell-e azt továbbítania. Ha a kérés egy olyan szolgáltatást igényel, amely egy másik gépen fut, akkor az átirányító továbbítja a kérést a hálózatra (lásd a 7.3 ábrát).

Az átirányító általános megoldást kínál a szolgáltatások eléréséhez: segítségével a felhasználó úgy láthatja a hálózat erőforrásait, mintha azok a helyi környezet részét képeznék. Egy távoli gépen működő merevlemez például ugyanúgy használható, mint a helyi gép merevlemeze.



7.3 ábra
Az átirányító

Webszolgáltatások

A HTTP (Hypertext Transfer Protocol) is az alkalmazási rétegben működik – ez teszi lehetővé a világháló (World Wide Web) működését. A HTTP-t eredetileg szövegek és képek továbbítására szánták, de a webszolgáltatások modelljének fejlődése lehetőséget teremtett számos más hasonló protokoll és alkotóelem kialakulására, amelyekkel sokféle, webböngészőben használható eszköz kialakult. A 20. órában további részleteket is megtanulunk a HTTP-ről és a webszolgáltatások módszereiről.

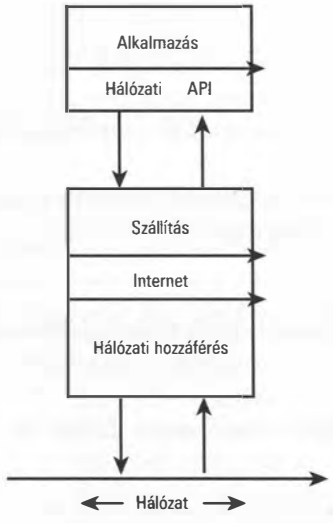
Alkalmazásprogramozási felületek az alkalmazási rétegben

Alkalmazásprogramozási felületnek (API-nak) hívjuk bizonyos függvények előre definiált gyűjteményét, amelyek révén egy program hozzá tud férni működési környezetének bizonyos részeihez. A különféle programok API függvényeket használnak az operációs rendszerrel való kommunikációhoz. Az API elvének egyik legtipikusabb megvalósítása a hálózati protokollverem. Ahogy az a 7.4 ábrán is látszik, a hálózati API egy alkalmazás és a protokollverem között létesít kommunikációs felületet. Az alkalmazói program az API függvényeit használja, hogy kapcsolatot nyisson (vagy bezárjon), és adatokat írjon/olvasson a hálózaton keresztül.

A foglalatkezelő (*Sockets*) API-t eredetileg a BSD Unixhoz tervezték, hogy az alkalmazások számára felületet biztosítson a TCP/IP protokollverem eléréséhez.
A foglalatkezelő manapság már széles körben elterjedt a különféle operációs rendszerek között, mint a TCP/IP egyik programozási felülete. Néhány éve a Microsoft is létrehozott (WinSock néven) egy saját foglalatkezelő változatot. A Windows-felhasználók-

nak a TCP/IP hálózat eléréséhez külön telepíteniük kellett (és be kellett állítaniuk) a WinSock-ot a Windows 3.1 verziójáig. A Windows 95-től kezdve a Microsoft már az operációs rendszerbe építi be a TCP/IP programozási felületét.

A foglalatkezelő API-hoz hasonló hálózati API-k foglalatokon (*socket*; lásd a 6. órát) keresztül fogadják és küldik az adatokat az alkalmazásnak. Ezek az API-k mind az alkalmazási réteghez tartoznak.



7.4 ábra
A hálózati API teszi lehetővé az alkalmazói programok számára, hogy TCP/IP-n keresztül hozzáférjenek a hálózathoz.

TCP/IP segédprogramok

Az alkalmazási réteg résztvevői közé tartoznak a TCP/IP segédprogramok is (7.2 ábra). A TCP/IP segédprogramok eredetileg az Internet és a korai Unix hálózatok számára készültek. Manapság ezeket széles körben használják a TCP/IP hálózat beállításához, kezeléséhez és a hibák megkereséséhez, és szinte minden operációs rendszerhez elérhetőek.

7.2 táblázat *TCP/IP segédprogramok*

Segédprogram	Leírása
Összeköttetéssel kapcsolatos segédprogramok	
`IPConfig`	Windows-os segédprogram, amely megjeleníti a TCP/IP beállításokat. Unix/Linux operációs rendszeren erre a célra az `ifconfig` használható.
`Ping`	A hálózati összeköttetés meglétét vizsgáló segédprogram.

7.2 táblázat *TCP/IP segédprogramok* *(folytatás)*

Segédprogram	Leírása
`Arp`	A helyi vagy egy távoli gép ARP gyorstárát megmutató (vagy módosító) segédprogram. Az ARP gyorstárban van eltárolva az, hogy milyen fizikai címek tartoznak az egyes IP-címekhez (lásd a 4. órát, „Az internet réteg").
`Traceroute`	A világhálón haladó adatcsomag útvonalát nyomon követő segédprogram.
`Route`	Az útválasztó tábla megtekintését vagy módosítását lehetővé tevő segédprogram (lásd a 8. órát, „Útválasztás").
`Netstat`	Az IP, az UDP, a TCP és az ICMP protokollok statisztikáit megjelenítő segédprogram.
`NBTstat`	A NetBIOS és az NBT statisztikáit megjelenítő segédprogram.
`Hostname`	A helyi gép nevét visszaadó segédprogram.
Fájlátvitellel kapcsolatos segédprogramok	
`Ftp`	TCP-t használó alapvető fájlátviteli segédprogram.
`Tftp`	UDP-t használó alapvető fájlátviteli segédprogram. `Tftp`-t szoktak használni a hálózati eszközök programkódjának letöltéséhez.
`Rcp`	Egyszerű fájlátviteli segédprogram távoli gépek fájljainak mozgatására.
Távoli eléréssel kapcsolatos segédprogramok	
`Telnet`	A legegyszerűbb terminálprogram.
`Rexec`	Távoli számítógépeken (az ott futó `rexecd` háttérprogram segítségével) parancsok végrehajtását lehetővé tevő segédprogram.
`Rsh`	Távoli számítógépeken parancshéj (*shell*) meghívása révén parancsvégrehajtást lehetővé tevő segédprogram.
`Finger`	Felhasználói információt megjelenítő segédprogram.
Internethasználati segédprogramok	
Böngészők	A világhálón található HTML (és más) tartalmakhoz történő hozzáférést lehetővé tevő segédprogramok.
Hírolvasók	Az internetes hírcsoportokhoz történő kapcsolódást lehetővé tevő segédprogramok.
Email ügyfélprogramok	Email-ek küldését és fogadását lehetővé tevő segédprogramok
Archie	Egykor népszerű internetes segédprogram, amely révén lehetőség nyílik a névtelenül használható (*anonymous*) FTP helyek tartalmának böngészésére. A világháló (World Wide Web) elterjedése és a keresőmotorok kialakulása visszaszorította az *Archie* fontosságát.

7.2 táblázat *TCP/IP segédprogramok* *(folytatás)*

Segédprogram	Leírása
Gopher	Menüvezérelt internetes információszerző segédprogram, amely – a világháló (World Wide Web) elterjedésével – népszerűségét illetően az Achie sorsára jutott.
`Whois`	Személyes elérhetőségi adatokat tartalmazó címtárhoz hozzáférést nyújtó segédprogram, afféle internetes telefonkönyv.

Összefoglalás

Ezen az órán a TCP/IP alkalmazási rétegéről volt szó, valamint az itt működő alkalmazásokról és szolgáltatásokról. Néhány TCP/IP segédprogramról is említést tettünk.

Kérdések és válaszok

K *Egy fájlkiszolgálóként működő számítógép rá van kötve a hálózatra, ám a felhasználók mégsem érik el az ott található állományokat. Mi van rosszul beállítva?*

V Számos dologgal probléma lehet. Ilyen esetben alaposabban meg kell vizsgálni a szóban forgó operációs rendszert és a beállításokat. Ennek az órának az ismereteire építve annyit mindenképpen érdemes megnézni, hogy el van-e indítva a fájlkiszolgálási szolgáltatás a szóban forgó gépen. A fájlkiszolgáló nem pusztán egy (esetleg erre a célra) kijelölt számítógép, hanem egy szolgáltatás, amely egy (általában erre a célra üzemeltetett) gépen fut.

K *Az OSI modell miért különíti el az alkalmazási réteg függvényeit három különböző (viszonylati, megjelenési és alkalmazási) réteggé?*

V Az alkalmazási réteg számos különböző szolgáltatást nyújt. Az OSI modell által megadott alrétegek segíthetnek a szoftverfejlesztőknek, hogy moduláris szerkezetben alakítsák ki a program egyes alkotóelemeit. Az alkalmazásfejlesztőket is segítheti a finomabb rétegekre bontás: így még célszerűbb felületeket alakíthatnak ki a protokollverem eléréséhez.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **Fájlszolgáltatás** (*File service*) – Olyan szolgáltatás, amely hálózati adattárolóknak küldött írási és olvasási kéréseket teljesít.
- **Névfeloldási szolgáltatás** (*Name resolution service*) – Olyan szolgáltatás, amely a hálózati IP-címekhez felhasználóbarát neveket rendel.
- **Nyomtatási szolgáltatás** (*Print service*) – Olyan szolgáltatás, amely hálózati nyomtatókra küldött dokumentum-nyomtatási kéréseket teljesít.
- **Átirányító** (*Redirector*) – Olyan szolgáltatás, amely a helyi gépen kiadott erőforrás-használati igényről eldönti, hogy kiszolgálható-e helyben, vagy csak a hálózaton keresztül.
- **Foglalatkezelő** (*Sockets*) – Eredetileg a BSD Unixhoz kialakított hálózati API, amely a TCP/IP-hez hozzáférést biztosító alkalmazások számára készült.

III. RÉSZ

A hálózat használata TCP/IP segítségével

8. ÓRA

Útválasztás

Ebben az órában a következőkről lesz szó:

- IP továbbítás
- Közvetlen és közvetett útválasztás
- Útválasztó protokollok

Az internetet (vagy bármilyen globális hálózatot) támogató infrastruktúra nem működhet útválasztók (*routers*) nélkül. A TCP/IP-t arra tervezték, hogy útválasztók használatával működjön, így a TCP/IP tárgyalásából nem maradhat ki az útválasztók működésének végiggondolása. Ezen az órán megtanuljuk, hogy milyen összetett kommunikációs folyamat zajlik az útválasztók között a hálózaton annak érdekében, hogy meghatározzák a csomagok haladásának optimális útvonalát. Ezen az órán tehát útválasztókról, útválasztó táblákról és útválasztó protokollokról lesz szó.

Az óra anyagának elsajátítása után az olvasó képes lesz

- jellemezni az IP továbbítás folyamatát
- különbséget tenni a távolságvektor és a kapcsolatállapot alapú útválasztás között
- vázolni a gerinc (*core*), a belső (*interior*) és a külső (*exterior*) útválasztók szerepét
- jellemezni a két leggyakoribb útválasztó protokollt, a RIP-et (útválasztó információs protokoll) és az OSPF-et (nyitott legrövidebb út protokoll)

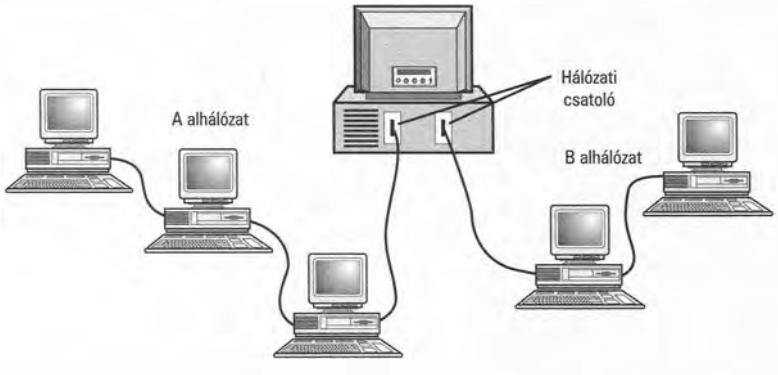
Útválasztás TCP/IP hálózatban

Eredeti formájában az útválasztó egy olyan eszköz, amely logikai cím alapján irányítja a forgalmat. Egy hagyományos hálózati útválasztó az internet rétegben (az OSI modell szerint a hálózati rétegben) működik, és az (internet réteg fejlécében található) IP-cím információk alapján dolgozik. Az OSI modell szóhasználatával élve (ahol a hálózati réteg a 3. réteg) az útválasztót „3. rétegbeli eszköz"-nek is hívják. Az elmúlt években a hardvergyártók olyan útválasztókat fejlesztettek ki, amelyek az OSI protokollverem magasabb szintű rétegeivel is együttműködnek. Ezen az órán a 4-től 7-ig terjedő rétegbeli útválasztókról is szó esik majd, de egyelőre fogjuk fel úgy az útválasztókat, mint olyan eszközöket, amelyek az OSI 3. („internet") rétegében működnek – ott, ahol az IP címzésnek kulcsszerepe van.

Az útválasztók alapvető fontosságúak minden nagyobb TCP/IP hálózatban. Útválasztók (és a TCP/IP útválasztó protokollok) nélkül az internet nem működhetne, és nem is nőtt volna ekkora méretűvé, mint amekkora manapság.

Az internet (és bármilyen nagyobb hálózat) számos útválasztót tartalmazhat, így ezek révén egy-egy adatcsomag többféle útvonalon is haladhat kiindulópontja és rendeltetési célja között. Az útválasztóknak egymástól függetlenül kell működniük, ám a működésük végeredménye nem lehet más, mint hogy az adatcsomagok pontosan és hatékonyan célba érjenek a világhálón.

Az adatcsomagok egyik hálózatról a másikra történő átjuttatásakor az útválasztók lecserélik a hálózat-elérési rétegfejléc információt, így össze tudnak kötni eltérő hálózattípusokat is. Vannak olyan útválasztók is, amelyek részletes információkat tárolnak a legjobb útvonalakról – ezeket távolság-, sávszélesség- és időadatok alapján állapítják meg. Az útvonalválasztó felderítési protokollokról még továbbiakat is fogunk tanulni ezen az órán.



8.1 ábra
Útválasztóként működő többkapcsolatú számítógép

A TCP/IP útválasztás olyan téma, amely (a könyv írásakor) 162 különböző RFC-ben (szabványjavaslatban) kerül említésre, és amelyről egy tucatnyi könyvet lehetne írni. Ami igazán fontos a TCP/IP útválasztással kapcsolatban, az az, hogy jól működik. Egy hétköznapi felhasználó el tudja indítani a böngészőjét, és csatlakozni tud számítógépével a kiválasztott kínai vagy finn webkiszolgálóhoz anélkül, hogy törnie kellene a fejét, hogy milyen hardvereszközökön jutnak keresztül adatcsomagjai szerte a világban. Kisebb hálózatokban is elmondható, hogy az útválasztás kulcsszerepet játszik a forgalom irányításában és a hálózat megfelelő sebességének megőrzésében.

Mi az az útválasztó?

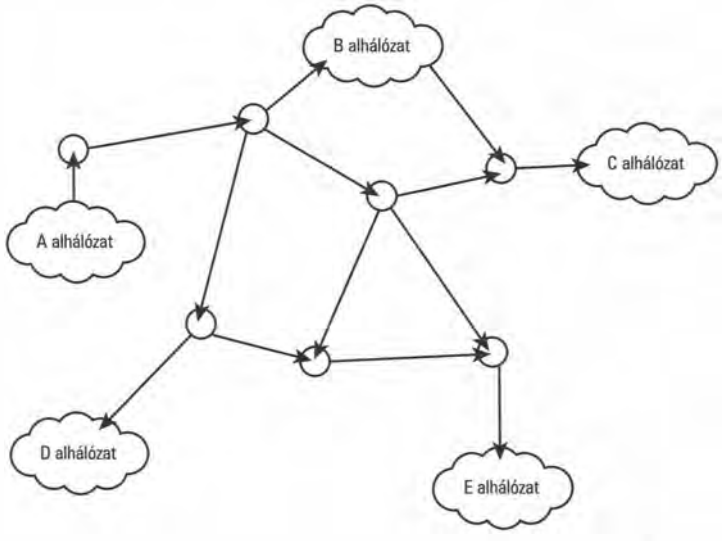
Talán a külalakjának leírásával lehet a legjobban jellemezni egy útválasztót. A legegyszerűbb (vagy legalábbis a legalapvetőbb) formájában az útválasztó egy olyan számítógép, amelynek két hálózati csatolója van. Az első útválasztók ténylegesen ilyenek voltak; **többkapcsolatú** (*multihomed*) számítógépeknek hívták őket. A 8.1 ábra mutat egy ilyen többkapcsolatú számítógépet, amely útválasztóként üzemel.

Az útvonalválasztás megértéséhez először azt kell tudatosítanunk, hogy IP-cím egy hálózati csatolóhoz (*adapter*) tartozhat, nem pedig egy számítógéphez. A 8.1 ábrán látható középső számítógéphez két IP-cím is tartozik (csatolónként egy-egy). Megoldható, hogy a két csatoló (fizikailag is) teljesen különböző IP alhálózatokhoz kapcsolódjon (ahogy az a 8.1 ábrán is látszik). A 8.1 ábrán látható többkapcsolatú számítógép protokollszoftvere az A alhálózatból is tud adatokat fogadni, ellenőrizni tudja az IP-cím információt, hogy megvizsgálja, hogy az adatoknak a B alhálózatba kell-e eljuttatni; ennek megfelelően le tudja cserélni a hálózat-elérési rétegfejléc adatokat a B alhálózatra vonatkozó fizikai címadatokra, majd pedig tovább tudja küldeni az adatokat a B alhálózatba. Ebben a helyzetben a többkapcsolatú számítógép útválasztóként működik.

Ha meg akarjuk érteni a hálózatok működését, képzeljük el az imént vázolt helyzetre épülően az alábbi bonyodalmakat:

- Az útválasztónak több, mint két hálózati csatolója is lehet, és így kettőnél több alhálózattal is tud kommunikálni. Egyre bonyolultabb lesz eldönteni, hogy hova kell továbbítani egy adatcsomagot, és a választható alternatív útvonalak száma is megnő.
- Az útválasztók által összekötött alhálózatok más alhálózatokkal is összeköttetésben állnak. Más szóval: az útválasztó olyan hálózati címeket is lát, amely alhálózatokkal nincs közvetlenül összekötve. Az útválasztóknak rendelkezniük kell stratégiával arra vonatkozóan is, hogy hogyan továbbítsanak csomagokat olyan alhálózatokba, amelyekkel nincs közvetlen összeköttetésük.
- Az útválasztók hálózata több alternatív útvonalat is lehetővé tesz – minden útválasztónak tudnia kell, hogy mi alapján döntsön a különböző útvonal-lehetőségek közül.

Ha a 8.1 ábrán látható egyszerű helyzetet gondolatban így fejlesztjük tovább, akkor már közelebb járunk a valódi útválasztók feladataihoz (8.2 ábra).



8.2 ábra
Útválasztás egy bonyolult hálózatban

A mai útválasztókat nem „többkapcsolatú számítógépekként" érdemes elképzelnünk. Költséghatékonyabbnak bizonyult speciális célhardvereket kifejleszteni erre a feladatra. Az útválasztókat arra optimalizálják, hogy a forgalomterelés feladatát hatékonyan el tudják látni; nem rendelkeznek az általános célú számítógépek minden tulajdonságával.

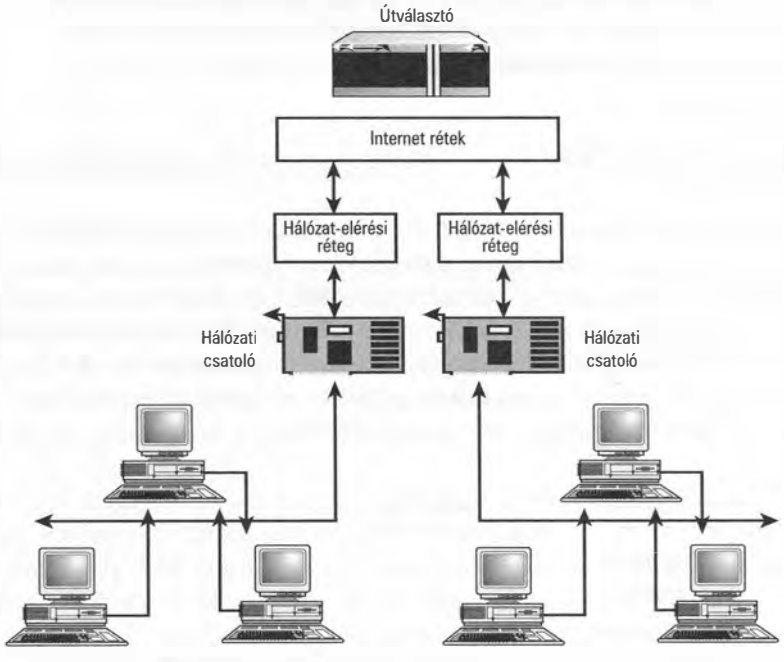
Az útválasztás folyamata

Az előző szakaszban vázolt egyszerű útválasztó fogalmára építve megfogalmazhatjuk a valódi útválasztók általános feladatait:

1. Az útválasztó adatcsomagokat fogad valamelyik alhálózatból (azok közül, amelyekhez kapcsolódik).
2. Az útválasztó elküldi az adatokat a protokollverembe az internet rétegnek. Más szóval: az útválasztó eldobja a hálózat-elérési rétegfejléc információkat, és (ha kell) új IP adatcsomagot készít.
3. Az útválasztó megvizsgálja az IP fejlécben található rendeltetési címet. Ha ez abban az alhálózatban található, ahonnan a csomag érkezett, akkor az útválasztó eldobja az adatcsomagot. (Ilyenkor ugyanis az adatcsomag már valószínűleg célba ért, hiszen olyan hálózatból lett továbbítva, amelyben a célgép megtalálható.)

4. Ha a rendeltetési cím egy másik alhálózatba mutat, akkor az útválasztó megnézi az útválasztó táblát, amiből kiderül, hogy merre kell továbbítania az adatcsomagot.
5. Miután az útválasztó eldöntötte, hogy melyik csatolójára fogja küldeni az adatokat, átadja az adatcsomagot a megfelelő hálózat-elérési réteg szoftverének, hogy küldje ki a kijelölt csatolón keresztül a hálózatra.

Az útválasztás folyamatát mutatja a 8.3 ábra. A 4. pontban említett útválasztó tábla gyakran kulcsszerepet játszik a folyamatban. Ennek az útválasztó táblának a kialakítására szolgáló protokoll jól jellemzi magát az útválasztót. Az útválasztókkal kapcsolatos viták többsége arról szól, hogy hogyan érdemes felépíteni az útválasztó táblát; valamint arról, hogy a táblák felépítéséhez szükséges információkat összegyűjtő útválasztási protokollok mennyiben visznek közelebb ahhoz, hogy az útválasztók hálózata egységes rendszerként működjön.



8.3 ábra
Az útválasztás folyamata

Az útválasztó táblák kialakítására szolgáló módszereket két alapvető csoportba sorolhatjuk aszerint, hogy milyen információkat használunk:

- **Statikus útválasztás** – A hálózati adminisztrátornak kézileg kell megadnia az útválasztási információkat.
- **Dinamikus útválasztás** – Az útválasztó táblához szükséges információkat az útválasztó protokollok számítják ki dinamikus módon, működés közben.

A statikus útválasztás hasznos lehet bizonyos esetekben. Azonban – ahogy azt bizonyára sejti is az olvasó – egy olyan rendszernek, amely manuális hálózatszervezési beállításokat igényel, komoly korlátai vannak. Egyrészt, a statikus útválasztást nem lehet igazán jól behangolni nagyméretű hálózatokon, ahol akár útválasztók százai üzemelnek. Másrészt, még a kisebb hálózatokon is igaz az, hogy a statikus útválasztáshoz szükséges adminisztráció aránytalanul sok időráfordítást igényel a hálózati adminisztrátortól, akinek nemcsak egyszer kell ezeket az adatokat megadnia, hanem a folyamatos frissen tartásról is gondoskodnia kell. Harmadrészt, egy statikus útválasztó nem tud olyan gyorsan alkalmazkodni a hálózat változásaihoz, mint ahogy egy megokosított (dinamikus) útválasztó tud.

A legtöbb dinamikus útválasztó megadja a lehetőséget az adminisztrátornak arra, hogy felülbírálja a dinamikusan kiválasztott útvonalakat, és bizonyos címekhez statikus haladási utakat adhasson meg. Az előre beállított statikus útválasztókat igen jól lehet használni hálózati hibakereséshez. A hálózati adminisztrátor bizonyos esetekben amiatt is megadhat egy-egy statikus haladási útvonalat, mert tudja, hogy egy adott irány mentén gyors hálózati eszközök vannak, vagy hogy ezzel kiegyensúlyozza az aránytalan hálózati forgalmat.

Az útválasztó táblák elmélete

Az útválasztó táblák (és más, internet rétegbeli forgalomirányító elemek) szerepe abban áll, hogy az adatcsomagok eljussanak a megfelelő helyi hálózatba. Ha egy adatcsomag eljutott a helyi hálózatba, akkor a hálózat-elérési protokollok dolga lesz a pontos kézbesítés. Az útválasztó tábláknak így nem kell teljes IP-címeket ismerniük, hanem elegendő hálózat-azonosítók szerint nyilvántartaniuk a címeket. (Lásd a 4. órát: „Az internet réteg" és az 5. órát: „Alhálózatok és a CIDR" – itt került szóba az, hogy az IP-címből hogyan derül ki a gép és a hálózat azonosítója.)

Rendeltetési hely →	A „következő lépés" →	Útválasztó kapu →
129.14.0.0	közvetlen kapcsolat	1
150.27.0.0	131.100.18.6	3
155.111.0.0	közvetlen kapcsolat	2
165.48.0.0	129.14.16.1	1

8.4 ábra
Útválasztó tábla

Egy végletekig egyszerűsített útválasztó tábla tartalmát láthatjuk a 8.4 ábrán. Egy útválasztó tábla alapvetően arra szolgál, hogy a rendeltetési hely hálózat-azonosítóját összekapcsolja a „következő lépés" (*next hop*) IP-címével. Ez az a hely, ahová az adatcsomagnak (rendeltetési helyére vezető útja során) a következő lépésben el kell jutnia. Figyeljük meg, hogy az útválasztó tábla megkülönbözteti az útválasztóval közvetlenül kapcsolatban lévő alhálózatokat azoktól, amelyek csak más útválasztók közvetítésével érhetőek el. A „következő lépés" lehet a rendeltetési hely alhálózata (ha az közvetlenül

elérhető), vagy az adatcsomag haladási irányába eső következő útválasztó. A 8.4 ábrán látható útválasztó kapu (*Router Port Interface*) utal az útválasztónak azon kapujára („lábára"), amelyen át az adott esetben továbbítania kell az adatokat.

Az útválasztó tábla „következő lépés" (*next hop*) oszlopának megértése elkerülhetetlen ahhoz, hogy legyen fogalmunk a dinamikus útválasztásról. Egy összetett hálózatban többféle alternatív útvonal is vezethet egy adatcsomag rendeltetési helyére, és az útválasztónak döntenie kell, hogy a következő lépésben ezek közül melyik felé irányítsa az adatokat. A dinamikus útválasztó ezt a döntést *útválasztó protokollok* révén megszerzett információk alapján hozza meg.

Minden számítógépnek lehet olyasféle útválasztó táblája, mint amilyen egy „igazi" útválasztónak is van. Mivel azonban egy normál számítógép általában nem végez útválasztási feladatokat, ez a tábla igen egyszerű. Az ilyen gépeken is meg szoktak adni egy „alapértelmezett útválasztót" vagy „alapértelmezett átjárót" (*default gateway*). Ez az az útválasztó, amely akkor kapja meg az adatcsomagot, ha az semelyik helyi gépnek (sem más útválasztónak) nem kézbesíthető.

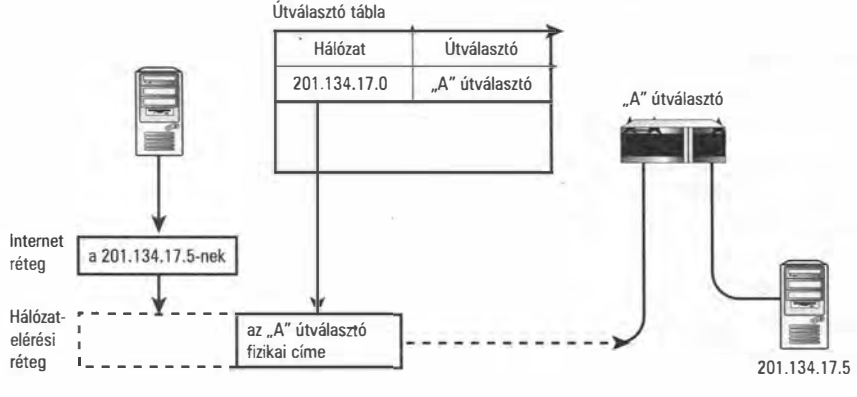
Egy pillantás az IP továbbításra

A normál számítógépeknek és az útválasztóknak is van útválasztó táblájuk. Egy normál számítógép táblája általában sokkal egyszerűbb, mint egy útválasztó gép táblája. Általában csak két sorból áll: az egyik a helyi hálózatra vonatkozik, a másik pedig az alapértelmezett átjáróra, ahová a helyi alhálózaton kézbesíthetetlen adatcsomagok kerülnek. Ez a két elemi forgalomirányítási információ elegendő ahhoz, hogy az adatcsomagok „tudják", merre kell haladniuk. Óránk során hamarosan megismerjük, hogy az útválasztók szerepe egy kicsivel bonyolultabb.

Ahogy megtanultuk a 4. órán, a TCP/IP program ARP-t használ az IP-cím (helyi hálózaton értelmezett) fizikai címmé történő alakításához. De mi történik akkor, ha az IP-cím nem a helyi hálózathoz tartozik? Ahogy a 4. órán szó volt róla: ilyenkor az útválasztóhoz küldjük az adatcsomagot. Bizonyára sejti az olvasó, hogy a valódi helyzet ennél egy fokkal bonyolultabb. A (4.3 ábrán bemutatott) IP fejléc csak a forráshely és a rendeltetési hely IP-címét tartalmazza. A fejlécben nincs annyi hely, hogy minden közbenső útválasztó címét eltárolja, amelyet az adatcsomag (célja felé vezető útja során) érint. Ahogy ezen az órán már elhangzott: az IP továbbítási folyamat során az útválasztó címe nem kerül be az IP fejlécbe; ehelyett a gazdagép mind az adatcsomagot, mind az útválasztó IP-címét leküldi a hálózat-elérési rétegbe. Itt a protokollszoftver újabb lekérdezést végez annak érdekében, hogy az adatcsomagot beillessze egy olyan keretbe (*frame*), amely az útválasztóhoz halad, helyi kézbesítéssel. Azaz a továbbított adatcsomag IP-címe arra a gépre mutat, amelyik ténylegesen meg fogja kapni az adatokat. Annak a keretnek a fizikai címe, amelyik az adatcsomagot egy útválasztónak továbbítja a helyi hálózaton: az útválasztó helyi csatolójának a címe.

Lássuk ennek a folyamatnak a rövid leírását (8.5 ábra).

1. Egy gép IP adatcsomagot akar küldeni egy másik gépnek; ehhez megvizsgálja a saját útválasztó tábláját.
2. Ha az adatcsomag kézbesíthetetlen a helyi hálózaton, akkor a gép (az útválasztó táblából) kiolvassa a csomag rendeltetési helyéhez tartozó útválasztó IP-címét. (Ha a gép egy helyi alhálózaton van, akkor ez az útválasztó IP-cím a legtöbbször magának az alapértelmezett átjárónak a címe.) Az útválasztó IP-címét az ARP protokoll alakítja át fizikai címmé.
3. A távoli gépnek címzett adatcsomagot megkapja a hálózat-elérési réteg. Hasonlóképp annak az útválasztónak a fizikai címét is, amelynek feladata lesz az adatcsomag továbbítása.
4. A keret megérkezik az útválasztó hálózati csatolójához, mivel a keret rendeltetési helyének fizikai címe megegyezik az útválasztó fizikai címével.
5. Az útválasztó kicsomagolja a keretet, majd az adatcsomagot felküldi az internet rétegbe.
6. Az útválasztó ellenőrzi az adatcsomag IP-címét. Ha ez megegyezik a saját IP-címével, akkor az adatot magának az útválasztónak szánták. Ha nem egyezik meg, akkor megpróbálja továbbítani az adatcsomagot. Ehhez megvizsgálja a saját útválasztó tábláját, hogy olyan útvonalat találjon, amely az adatcsomag rendeltetési helyéhez tartozik.
7. Ha az adatcsomag az útválasztóhoz közvetlenül csatlakozó valamennyi alhálózaton kézbesíthetetlen, akkor az útválasztó elküldi az adatcsomagot egy másik útválasztónak, és a folyamat (az 1. lépéstől kezdve) kezdődik elölről. Mindez így megy egészen addig, amíg az utolsó útválasztó már kézbesíteni tudja az adatcsomagot a rendeltetési helyére.



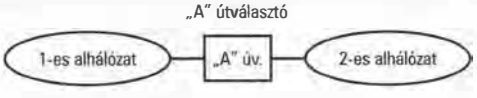
8.5 ábra

Az IP továbbítás folyamata

Az IP továbbítás folyamatának 6. lépése kulcsfontosságú. Az útválasztóra erősen jellemző, hogy ezt hogyan oldja meg. Ne feledjük: attól még nem lesz egy számítógép igazi útválasztó, hogy egynél több hálózati kártyája van. Ha a gépen nem fut az IP továbbításhoz szükséges program, akkor az adatcsomagok nem jutnak el az egyik csatolótól a másikig. Ha egy számítógép nincs felkészítve arra, hogy kezelni tudjon egy olyan adatcsomaggal, amelyet nem neki szántak, akkor egyszerűen eldobja ezt az adatcsomagot.

Közvetlen és közvetett útválasztás

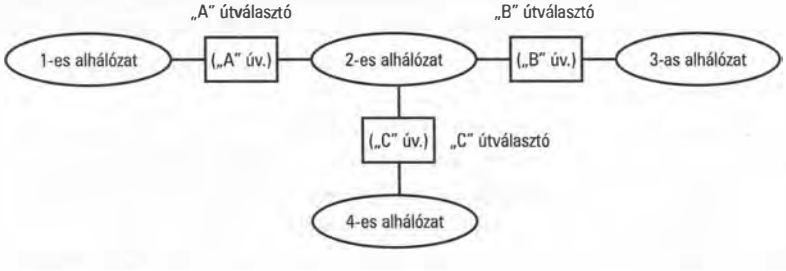
Ha egy útválasztónak csak két alhálózatot kell ellátnia, akkor az útválasztó tábla egyszerű is lehet. A 8.6 ábrán látható útválasztó sohasem találkozik olyan IP-címmel, amely ne tartozna valamelyik kapujához. Az ilyen útválasztó közvetlenül kapcsolódik minden alhálózatához. Más szóval, a 8.6 ábra útválasztója minden adatcsomagot közvetlen útválasztással tud kézbesíteni.



8.6 ábra

Ha egy útválasztó csak két alhálózatot kapcsol össze, akkor közvetlenül éri el mindkettőt.

Képzeljünk el egy fokkal bonyolultabb hálózatot (8.7 ábra), amelyben az „A" útválasztó nincs közvetlen összeköttetésben a 3-as alhálózattal, és (segítség nélkül) mit sem tud a 3-as alhálózatról. Ezt nevezzük **közvetett** útválasztásnak. A legtöbb (útválasztókkal megszervezett) hálózat bizonyos mértékig támaszkodik a közvetett útválasztásra. A nagyobb szervezeteknek több tucat útválasztója is lehet, és ezek között elenyésző azok száma, amelyek közvetlenül kapcsolódnak az összes alhálózathoz. Az efféle hálózatokról bőven lesz szó ezen az órán. Most az a legfontosabb kérdésünk a 8.7 ábrával kapcsolatban, hogy miként tájékozódhat az „A" útválasztó a 3-as alhálózatról? Honnan tudhatja az „A" útválasztó, hogy a 3-as alhálózatba küldött adatcsomagokat a „B" vagy a „C" útválasztónak kell továbbítania?



8.7 ábra

Ha az útválasztó nem csatlakozik közvetlenül egy alhálózathoz, akkor közvetett útválasztással kell megoldania az oda címzett adatcsomagok továbbítását.

Az útválasztók két módon kaphatnak eligazítást a közvetett útválasztással kapcsolatos teendőkről: a rendszergazdától vagy egymástól.

Ez a két lehetőség megfelel a statikus és a dinamikus útválasztásnak. A rendszergazda közvetlenül az útválasztó táblában is megadatja a hálózati útvonalakat (ez a statikus útválasztás), vagy (például) a „B" útválasztó is informálhatja az „A" útválasztót a 3-as alhálózatról (ez pedig a **dinamikus útválasztás**). A dinamikus útválasztásnak vannak előnyei. Egyrészt nem igényel emberi beavatkozást. Másrészt érzékeli a hálózat változásait. Ha egy új alhálózatot csatolnak a „B" útválasztóhoz, akkor a „B" útválasztó informálhatja a változásról az „A" útválasztót.

Ahogy sejthető, a statikus útválasztás leginkább a kicsi, egyszerű és változatlan hálózatokban hatékony. Használata elfogadható lehet még a 8.7 ábrán bemutatott helyzetben is, ám az útválasztók számának növekedtével a statikus útválasztás egyre kevésbé működőképes. A lehetséges útvonalak száma ugrásszerűen nő az újabb alhálózatok kialakításakor, és ez rengeteg többletmunkát igényel a rendszergazdától. De még ennél is nagyobb baj az, hogy a nagy hálózatokban alkalmazott statikus útválasztás lecsökkentheti a hálózat sebességét és kiszámíthatatlan viselkedéshez vezethet. Előfordulhat, hogy végtelen ciklusba kerül egy adatcsomag, azaz ugyanazon az útvonalon halad végig újra meg újra, és sohasem ér célba.

Meg kell jegyeznünk, hogy a 8.7 ábrán bemutatott hálózatot még meg lehetne szervezni alapértelmezett átjárók megadásával. Így az „A" útválasztónak nem kell tudnia semmit a 3-as alhálózatról. Mindazon csomagokat, amelyekkel nem tud mit kezdeni, egyszerűen továbbküldi a „B" útválasztónak – döntse el ő, hogy mi a teendő. Az ilyen eljárás azonban nem működik sokkal nagyobb hálózatokban. Az alapértelmezett átjáró használata statikus útválasztást jelent, és ha magukat az útválasztókat is alapértelmezett átjárókra irányítjuk, akkor egy bonyolultabb hálózatban ez olyan problémákat fog okozni, mint amik a statikus útválasztással járnak együtt; rossz hatékonyságot és kiszámíthatatlan viselkedést eredményez.

Ilyen okok miatt a mai útválasztók inkább a dinamikus útválasztás valamilyen formáját alkalmazzák. Az útválasztók egymással kommunikálva megtudhatják, hogy mi a helyzet a távoli alhálózatokkal és a lehetséges haladási útvonalakkal. Ezek alapján minden útválasztó felépítheti a saját tábláját. A következő részekben arról lesz szó, hogy hogyan is működik konkrétan a dinamikus útválasztás.

A valódi útválasztók általában a statikus és dinamikus útválasztás valamilyen keverékét használják. A rendszergazda előírhat néhány statikus útvonalat, és lehetővé teheti, hogy a többit dinamikusan alakítsa ki az útválasztó. A statikusan beállított útvonalak hasznosak lehetnek akkor, ha egy adott forgalmi utat szeretnénk kijelölni. A rendszergazda (megfelelő útválasztó-beállításokkal) megadhatja például, hogy a forgalom egy nagy sávszélességű útvonalon haladjon.

Dinamikus útválasztási módszerek

Az egy csoportba tartozó útválasztók elegendő információt adhatnak egymásnak a hálózatról ahhoz, hogy minden útválasztó fel tudja építeni a saját tábláját. Ez alapján bármely alhálózatba címzett adatcsomagot megfelelően tovább tudnak küldeni. Pontosan miről is kommunikálnak azonban az útválasztók? Hogyan építik fel útválasztó tábláikat? Nyilván rájött már az olvasó, hogy egy útválasztó működése lényegében az útválasztó táblától függ. Többféle útválasztó protokoll is létezik. Ezek legtöbbje a két legalapvetőbb algoritmus egyikére épül: vagy a távolságvektor alapú, vagy a kapcsolatállapot alapú útválasztásra.

Ez a két módszer nem más, mint két különböző megközelítés az útválasztási információk összegyűjtésére és az útválasztók közti kommunikációra. Az alábbi szakaszokban a távolságvektor alapú (*distance vector*) és a kapcsolatállapot alapú (*link state*) útválasztásról lesz szó. Később közelebbről is megvizsgálunk két protokollt, amely ezekre a módszerekre épül: a RIP-et, amely egy távolságvektor alapú útválasztó protokoll, valamint az OSPF-et, amely egy kapcsolatállapot alapú protokoll.

A távolságvektor alapú (*distance vector*) és a kapcsolatállapot alapú (*link state*) útválasztási algoritmus igazából egy-egy *protokoll-osztályt* jelent. Az (ezek alapján) konkrétan megvalósított protokollok a fenti algoritmus jellemzőin kívül még egyéb funkciókkal és tulajdonságokkal is rendelkeznek. Számos útválasztóban beépített módon megtalálhatóak indító szkriptek, statikus útválasztási bejegyzések és más jellemzők, amelyek árnyalják a távolságvektor alapú és a kapcsolatállapot alapú útválasztás elméleti módszereit.

Távolságvektor alapú útválasztás

A **távolságvektor alapú útválasztás** (más néven a **Bellman-Ford útválasztás**) hatékony és egyszerű útválasztási módszer, amelyet számos útválasztó protokoll alkalmaz. Egy időben a távolságvektor alapú útválasztás volt a meghatározó megoldás a hálózati iparban. Ma is eléggé gyakori, bár az utóbbi időben a kifinomultabb módszerek miatt (mint amilyen például a kapcsolatállapot alapú útválasztás) kissé visszaesett a népszerűsége.

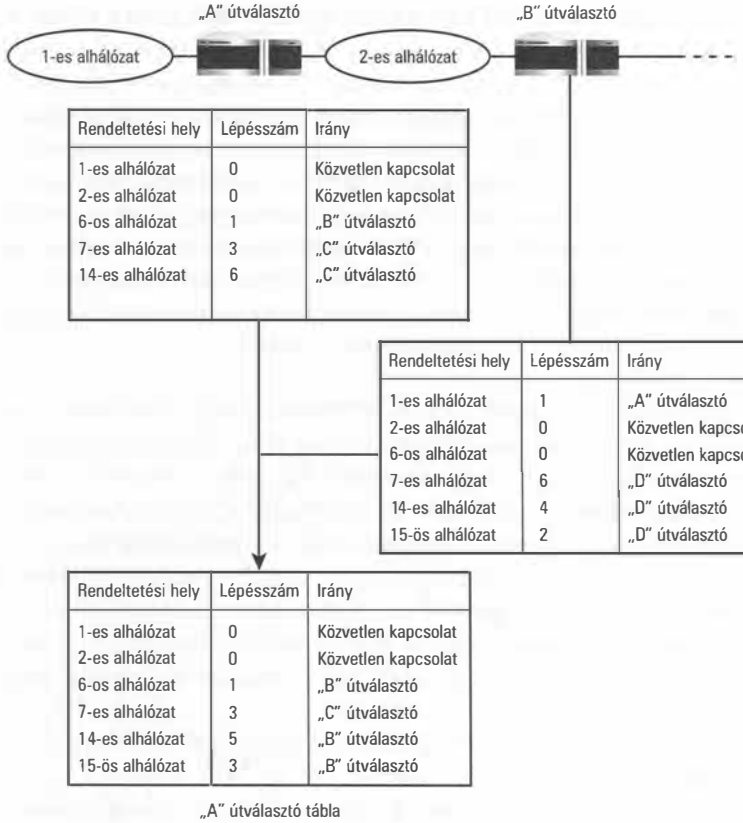
A távolságvektor alapú útválasztást arra tervezték, hogy az útválasztóknak a lehető legkevesebbet kelljen egymással kommunikálniuk, és hogy a lehető legkevesebb adatot kelljen tárolni az útválasztó táblában. A távolságvektor alapú útválasztás mögött az a filozófia húzódik, hogy az útválasztónak nem kell tudnia az összes alhálózathoz vezető utat – csak azt kell tudnia, hogy milyen irányban (innen a *vektor* elnevezés) kell elküldenie egy adatcsomagot, hogy eljusson egy adott alhálózathoz. Az alhálózatok közti távolságot úgy szokták mérni, hogy hány útválasztón kell áthaladnia az adatoknak, míg eljutnak egyik-

ből a másikba. Innen az elnevezés is: „lépésszám", *hop count*. A távolságvektoros algoritmust használó útválasztók ennek a távolságnak a minimalizálásával próbálják meg optimalizálni az adatcsomagok célba jutását (azaz: minél kevesebbet kelljen lépniük, minél kevesebb útválasztót kelljen érinteniük az adatcsomagoknak).

A távolságvektor alapú útválasztás a következőképpen működik:

1. Amikor az „A" útválasztót bekapcsolják, akkor érzékeli, hogy mely alhálózatokhoz kapcsolódik közvetlenül – ezeket elhelyezi az útválasztó táblájába. Ezekhez a közvetlenül kapcsolódó alhálózatokhoz 0 lépésszám tartozik, hiszen az ide tartó adatcsomagoknak nem kell más útválasztót érinteniük ahhoz, hogy elérjék a rendeltetési helyükként kitűzött alhálózatot.
2. Az útválasztók bizonyos időközönként jelentést kapnak a szomszédos útválasztóktól. Ebben a jelentésben az szerepel, hogy az adott útválasztók milyen alhálózatokról tudnak, és hogy mennyi az egyes alhálózatokhoz tartozó lépésszám.
3. Amikor az „A" útválasztó jelentést kap a szomszédos útválasztótól, akkor a friss forgalomirányítási információt beépíti a saját útválasztó táblájába, mégpedig az alábbi módon:
    - Ha a „B" útválasztó ismer olyan alhálózatot, amelyet az „A" útválasztó nem ismer (nincs a táblájában), akkor akkor felveszi ezt az adatot is a saját táblájába. Az ide vezető út nyilván „B" útválasztón keresztül vezet. Azaz, ha az „A" útválasztó olyan adatcsomagot kap, amelyet ebbe a (most megismert) új alhálózatba címeztek, akkor ezt a csomagot a „B" útválasztónak fogja elküldeni. Az ehhez az alhálózathoz tartozó lépésszám a „B"-től származó jelentésben szereplő lépésszámnál eggyel nagyobb lesz, hiszen az „A" útválasztó egy lépéssel távolabb van ettől alhálózattól, mint a „B".
    - Ha a „B" útválasztó olyan alhálózatról ad hírt, amelyet az „A" útválasztó már ismer (azaz szerepel a táblájában), akkor (a kapott lépésszám eggyel való növelése után) az „A" útválasztó összehasonlítja a saját eltárolt lépésszám-adatával a friss információt. Amelyik hatékonyabb (azaz a kisebb lépésszám tartozik hozzá), azt őrzi meg „A" a táblájában. Ha az új irány a hatékonyabb, akkor a szóban forgó alhálózatba tartó adatcsomagokat a „B" útválasztónak fogja ezentúl küldeni. Ha az új irány kevésbé hatékony, akkor továbbra is a régi információra fog támaszkodni az „A" útválasztó, és ejti a jelentésbeli friss információt.
    - Ha a B útválasztón át elérhető szegmenshez tartozó felülvizsgált lépésszám (a B útválasztótól kapott lépésszám + 1) nagyobb, mint az az érték, ami jelenleg az A útválasztó táblázatában szerepel, akkor a B-n keresztül vezető útvonalat a rendszer nem használja. Az A ilyenkor a továbbiakban is azt az útvonalat használja, ami eddig is szerepelt az adatai között.

Az útválasztóknak minden ilyen jelentés után tisztább képük lesz a hálózatról. A forgalomirányítási információk lassanként elterjednek az egész hálózaton. Ez még akkor is hasznos, ha semmi változás nem történik a hálózat szerkezetében: az útválasztók fokozatosan „kitapasztalják", hogy melyik a leghatékonyabb útvonal az egyes alhálózatokhoz.



8.8 ábra
A távolságvektor alapú útválasztás táblafrissítő lépése

A 8.8 ábrán láthatjuk, hogy hogyan zajlik a távolságvektor alapú útválasztás táblájának frissítése. Figyeljük meg, hogy az ábrázolt pillanatban már történtek frissítések, hiszen mind az „A", mind a „B" útválasztó tud olyan alhálózatokról, amelyekhez nem kapcsolódnak közvetlenül. Ebben az esetben a „B" útválasztó hatékonyabb útvonalakat ismer a 14-es alhálózathoz, mint az „A", így az „A" útválasztó felül fogja írni a tábláját: ezentúl a 14-es alhálózatba tartó adatcsomagokat a „B" útválasztóhoz fogja irányítani. A 7-es alhálózathoz az „A" útválasztónak van hatékonyabb útvonala, ezért a táblázatnak ez a sora változatlanul marad. (Bekerül egy új alhálózat is a táblába, a 15-ös.)

A 8.8 ábrán látható rendeltetési helyeket (1-es alhálózat, 2-es alhálózat stb.) elképzelhetjük teljes IP hálózatként vagy alhálózatként is, a szövegösszefüggéstől függően.

Kapcsolatállapot alapú útválasztás

A távolságvektor alapú útválasztás akkor működik jól, ha a fent definiált „lépésszám" valóban pontosan jellemzi egy útvonal hatékonyságát. Ez gyakran igaz; vannak azonban esetek, amikor ennél bonyolultabb a valóság. (Egy lassú összeköttetésen átvezető út tovább tarthat, mint egy nagy sebességű kapcsolatot használó irány, még akkor is, ha ez utóbbi esetén több útválasztót kell érinteniük az adatcsomagoknak.) A távolságvektor alapú útválasztás az útválasztók számának növekedtével egyre pontatlanabb. Minden útválasztónak karban kell tartania egy táblabejegyzést az összes célpontra vonatkozóan, és a táblabejegyzések csak irányokat és lépésszámokat tartalmaznak. Az útválasztó nem tudja hatékonyan felhasználni a hálózat szerkezetére vonatkozó egyéb ismereteit. Óriási irány- és lépésszám-táblázatoknak kell áthaladniuk az útválasztókon olyan esetekben is, amikor az adatok többsége nem használható. Az informatikusok felvetették a kérdést, hogy nem lenne-e jobb megoldás. Ebből alakult ki a kapcsolatállapot alapú útválasztás elve. A kapcsolatállapot alapú útválasztás ma már elsőbbséget élvez a távolságvektor alapú útválasztással szemben.

A kapcsolatállapot alapú útválasztás mögött az a filozófia rejlik, hogy végső soron minden útválasztó megpróbálja a maga számára felépíteni az aktuális hálózati topológiát. Az útválasztók rendszeres időközönként állapotjelentést küldenek a hálózatra. Ezek az állapotjelentések tartalmazzák, hogy mely más útválasztókkal van közvetlen kapcsolatban az adott útválasztó, és azt is, hogy ez a kapcsolat milyen (működik-e az adott összeköttetés). Az útválasztók a többi útválasztótól kapott állapotjelentéseket felhasználják ahhoz, hogy kiépítsék a saját szemszögükből látható hálózati topológiát. Amikor egy útválasztónak továbbítania kell egy adatcsomagot, akkor a rendelkezésére álló adatok alapján megpróbálja kiválasztani a legjobb útvonalat, ami a rendeltetési helyhez vezet.

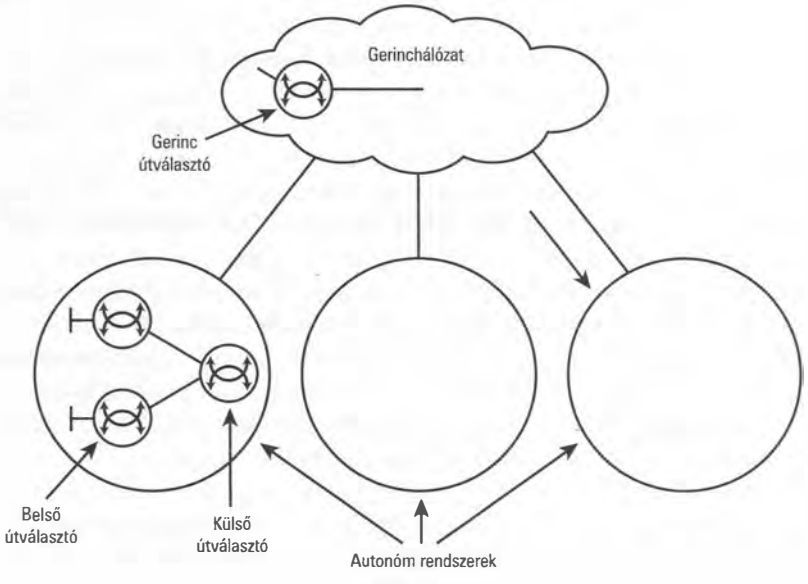
A kapcsolatállapot alapú útválasztás több feldolgozási időt igényel az egyes útválasztóktól, de a felhasznált sávszélesség lecsökken, mert az útválasztóknak nem kell a teljes táblát közzétenniük. Az esetleges hálózati problémákat is könnyebb megtalálni, mert egy adott útválasztótól származó állapotjelentés változatlanul halad végig a hálózaton. (Szemben a távolságvektor alapú útválasztással, amelynél a lépésszám-információ minden lépésben módosul.)

Útválasztás összetett hálózatokon

Mindeddig olyan eseteket vizsgáltunk, amely egyedi útválasztókról vagy útválasztók kisebb (de összefüggő) csoportjára vonatkozott. A valóságban vannak olyan nagy kiterjedésű hálózatok is, amelyekben útválasztók százai dolgoznak. A teljes interneten pedig útválasztók százezrei működnek. Az internethez hasonló nagy hálózatokon nem oldható meg, hogy az összes útválasztó a fent vázolt módon ossza meg információit a többi útválasztóval. Ha minden útválasztónak foglalkoznia kéne az összes többi útválasztótól kapott adattal, akkor az útválasztó protokoll forgalma (és az útválasztó táblák mérete)

hamarosan túllépne minden létező erőforrás-kapacitáson. Az internet szerencsére anélkül is működik, hogy minden útválasztó tudna minden más útválasztóról. Egy isztambuli fogorvos rendelőjében működő útválasztó évekig gond nélkül működhet anélkül, hogy bármi fogalma lenne a limai (Peru) festőművész műhelyének útválasztójáról. Ha a hálózatot hatékonyan megszervezték, akkor a legtöbb útválasztónak csak a tőle nem túl messze fekvő útválasztókkal kell információt cserélnie.

Az ARPAnet rendszerben, amely az internet őse volt, gerinc (*core*) útválasztók kis csoportja játszotta a központi gerinchálózat szerepét: önállóan működő egyedi hálózatokat kötöttek össze a segítségükkel. A gerinc útválasztók tudtak minden hálózatról, bár azon belül nem kellett tudniuk minden alhálózatról. Ha egy adatcsomag eljutott egy gerinc útválasztóig, akkor onnan már a rendszer bármely pontjára el tudott jutni. A gerincnek alárendelt hálózatokban működő útválasztóknak nem kellett tudniuk a világ összes hálózatáról – nekik csak annyit kellett megoldaniuk, hogy egymás között tudjanak adatokat küldeni, valamint legyen útvonaluk valamelyik gerinc útválasztóhoz.



8.9 ábra
Az internet útválasztóinak hierarchiája

Ebből fejlődött ki az a rendszer, amelynek sematikus képét a 8.9 ábrán láthatjuk. A gerinchálózat útválasztói közvetítik a különböző hálózatok közötti üzeneteket. A gerinchálózathoz önálló és független hálózatok kapcsolódnak (**autonóm rendszerek**). Egy autonóm rendszer lehet egy szervezeti hálózat, vagy (manapság gyakoribb módon) egy internetszolgáltató (ISP) hálózata. Az autonóm rendszer tulajdonosa maga intézheti a saját útválasztóinak beállítását. Az autonóm rendszerben található belső útválasztók meg tudják osztani egymással hálózatuk teljes topológiáját; viszonylag pontos útválasztó táblát alakíthatnak ki. Ha egy üzenet más hálózatnak van címezve, akkor azt a gerinc útválasztónak továbbítják. Fontos szerepet játszanak a külső (*exterior*) útválasztók is.

A **külső útválasztó** arra szolgál, hogy más hálózatokkal cseréljen információt. A teljes internet útválasztási adatforgalma ily módon alacsonyan tartható, mert csak a külső útválasztók küldenek útválasztási információkat a saját hálózatukon kívülre.

Többféle útválasztó létezik, és ezek különböző protokollokat és módszereket használnak az útválasztó tábla kialakításához. Az következő részekben még fogunk tanulni ezekről az útválasztó protokollokról. Az útválasztók az alábbi három alapvető típusba sorolhatóak:

- **Gerinc útválasztók** – A gerinc (*core*) útválasztók teljes körű információkkal rendelkeznek a többi gerinc útválasztóról. Az útválasztó táblájuk alapvetően annak a leképezése, hogy az autonóm rendszerek hol csatlakoznak be a gerinchálózatba. A gerinc útválasztók nem rendelkeznek részletes információkkal az autonóm hálózatokon belüli útvonalakról. A gerinchálózat útválasztási protokollja például a GGP (*Gateway-to-Gateway Protocol*), illetve elterjedőfélben van egy újabb, SPREAD-nek („*elterjeszt*") nevezett protokoll.
- **Külső útválasztók** – A külső (*exterior*) útválasztók nem tartoznak a gerinchálózathoz; ők csak az autonóm hálózatok között cserélnek forgalomirányítási információkat. Tisztában vannak a saját hálózatuk és a szomszédos autonóm hálózatok útválasztási adataival, de nem tárolják a teljes internetre vonatkozó szerkezeti információkat. A külső útválasztók régebben egy EGP-nek (*Exterior Gateway Protocol*) nevezett protokollt használtak; eredeti formájában ma ez már elavult. Ennek ellenére az új (külső útválasztók által használt) protokollokat is gyakran „EGP"-knek hívják. Ilyen például a BGP (*Border Gateway Protocol*). A külső útválasztók gyakran a saját autonóm rendszerük belső útválasztójaként is működnek.
- **Belső útválasztók** – Azokat az útválasztókat hívjuk belső (*interior*) útválasztóknak vagy **belső átjáróknak** (*gateway*), amelyek a saját autonóm területükön belül osztanak meg útválasztási információkat. Ezek az útválasztók egy IGP (*Interior Gateway Protocol*) nevű protokoll-osztályt használnak. Ilyen például a RIP (*Routing Information Protocol*) és az OSPF (*Open Shortest Path First*). Óránk során még részletesen visszatérünk a RIP-re és az OSPF-re.

Fontos tudnunk, hogy egy autonóm hálózaton belül is hierarchikus szerkezetbe lehet szervezni az útválasztókat. Egy nagy autonóm rendszer több csoportnyi belső útválasztót is tartalmazhat, amelyek között külső útválasztók cserélik ki az útválasztási információkat. Az autonóm hálózatok tulajdonosai szabadon megtervezhetik az útválasztók működési hierarchiáját és a használni kívánt útválasztó protokollt.

Az internet ma már annyira bonyolult, hogy ahhoz képest az itt vázolt ARPAnet gerinchálózati modell valójában csak leegyszerűsítés. Az internet gerincét általában egy áthatolhatatlan felhőnek szokták ábrázolni, amelybe itt-ott becsatlakoznak az autonóm hálózatok.

A belső útválasztók működése

Ahogy korábban megtanultuk, a belső útválasztók egy autonóm hálózat belső régiójában üzemelnek. A belső útválasztókban pontos információkat kell tárolni mindazon alhálózatokról, amelyek a (vele egy csoportban levő) többi útválasztóval összeköttetésben vannak. Nem kell foglalkozniuk azonban azokkal a hálózatokkal, amelyek az autonóm rendszeren kívül találhatóak.

Több belső útválasztó protokoll is létezik. A rendszergazdának ki kell választania a helyi hálózat szerkezetének és hardveres adottságainak leginkább megfelelőt. A következő szakaszokban a fontosabb belső útválasztó protokollokról lesz szó: a RIP-ről (*Routing Information Protocol*) és az OSPF-ről (*Open Shortest Path First*).

A RIP távolságvektor alapú útválasztó protokoll, az OSPF pedig kapcsolatállapot alapú. Mindkettőhöz tartoznak olyan részletek és problémamegoldó eljárások, amelyekről a korábban tárgyalt vázlatos leírásban nem esett szó.

A legtöbb mai útválasztó több protokollt is támogat.

RIP (Routing Information Protocol)

A RIP tehát egy távolságvektor alapú útválasztó protokoll, ami azt jelenti, hogy az alapján választja ki az adatcsomagok számára kijelölt utat, hogy melyikkel jár együtt a legkevesebb útválasztó érintése (azaz: merre lesz a legkisebb a „lépésszám"; lásd a „Távolságvektor alapú útválasztás" című korábbi szakaszt). A RIP-et a Berkeley-i Egyetemen (California) fejlesztették ki, és először a Berkeley Systems Design (BSD) Unix változatokban tett szert nagy népszerűségre. Aztán a RIP máshol is elterjedt, és igen kedvelt útválasztó protokollá vált. Ma is széles körben használják, bár bizonyos tekintetben ma már elavultnak számít. A RIP II. megjelenése valamelyest orvosolta azokat a problémákat, amelyek a RIP I. verzió kapcsán felmerültek. Ma számos útválasztó támogatja a RIP I. és RIP II. protokollt. Az IPv6 hálózatok számára kiterjesztett RIP II-t RIPng -nek hívják (*ng = new generation, új generációs*).

UNIX/Linux rendszereken futó `routed` háttérprogram RIP-et használ.

A RIP, mint tudjuk, távolságvektor alapú útválasztó protokoll, így azt igényli az útválasztóktól, hogy más útválasztókra figyelve információkat szerezzen az útvonalakról és lépésszámokról, és ezt építse be a saját táblájába. A RIP szereplői aktív és passzív

eszközök. Aktív RIP csomópontnak hívjuk azokat az útválasztókat, amelyek részt vesznek a távolságvektor-adatokra vonatkozó információcserében. Elküldik útválasztó tábláikat a többi útválasztónak és figyelik a tőlük érkező frissítéseket. A passzív RIP eszközök csak beépítik a frissítéseket, de nem publikálják a saját útválasztó táblájukat. Passzív RIP csomópontnak számítanak a normál hálózati számítógépek is. (Emlékezzünk vissza, hogy a normál hálózati számítógépeknek is van útválasztó táblája.)

Amikor korábban részletesen tárgyaltuk a távolságvektor alapú útválasztást, talán felmerült az olvasóban, hogy mi történik akkor, amikor a két összehasonlítandó lépésszám-adat éppen megegyezik. Ez olyan részlet, amelynek eldöntése az adott protokoll hatáskörébe tartozik. A RIP esetében ilyenkor (amikor két különböző útvonalhoz is ugyanolyan lépésszám tartozik) az eredetileg is a táblában levő változat marad meg. Ezzel nem alakul ki felesleges forgalomhullámzás, amely akkor jönne létre, ha az útválasztó folyamatosan cserélgetné táblája bejegyzéseit az azonos lépésszámú útvonalakra vonatkozóan.

A RIP útválasztók félpercenként publikálják frissített táblájukat. Lehetőségük van arra is, hogy kérjenek egy azonnali frissítést. A többi távolságvektor alapú útválasztó protokollokhoz hasonlóan a RIP is akkor működik a legjobban, ha a hálózat kiegyensúlyozott (az útválasztók számát illetően). Ha túl sok útválasztó van, akkor a táblák lassú átadása problémákhoz vezethet. Emiatt a RIP beállít egy maximumot arra vonatkozóan, hogy hány útválasztón haladhatnak át a táblák, míg a feladótól a címzettig megérkeznek. Ez a lépésszám-küszöbérték a RIP esetében 15. Ez a szám behatárolja a csoportban használható útválasztók számát, de ha az útválasztók hierarchikus felépítésben működnek, akkor egy nagy csoport is működtethető ezzel a 15-ös küszöbértékkel.

Bár a távolságvektor alapú útválasztás nem teszi lehetővé, hogy az adatátviteli sebességtől vagy a hálózat fizikai megvalósításától függjön a forgalomirányítás, az mégis megoldható, hogy a rendszergazda azáltal befolyásolja az útválasztást, hogy a kevéssé hatékony útvonalakhoz mesterségesen nagy lépésszám-adatot állít be.

A sebezhető RIP protokollt fokozatosan felváltják újabb protokollok, például az OSPF, amelyről a következő szakaszban olvashatunk.

OSPF (Open Shortest Path First)

Az OSPF egy naprakészebb útválasztó protokoll, amely fokozatosan kiszorítja a RIP-et. Az OSPF egy kapcsolatállapot alapú útválasztó protokoll. Először 1989-ben jelent meg az 1131-es RFC-ben (szabványjavaslatban; „*Request for Comment*"). Azóta többször is megváltozott. A 2328-as RFC az OSPF 2-es verzióját fedi le, és vannak még későbbi RFC-k is, amelyek az OSPF protokoll további kiterjesztéseit és alternatíváit taglalják. Az IPv6-ot is támogató, 3-as verziójú OSPF a 2740-es RFC-ben jelent meg.

Az OSPF útválasztócsoport útválasztói azonosítót kapnak. Az **útválasztó-azonosító** (általában) az útválasztó hatáskörébe tartozó, számszerűleg legnagyobb IP-cím. Ha az útválasztó visszacsatoló felületet (*loopback interface*) használ, akkor az útválasztó-azonosító a legnagyobb visszacsatolási cím. A visszacsatolási címekkel kapcsolatban lásd a 4. órát.)

Ahogy óránk során már megtanultuk, a kapcsolatállapot alapú módszernél az útválasztók a hálózati topológiát térképezik fel. Az útválasztók az útválasztó-azonosító alapján azonosítják egymást a topológián belül. Minden útválasztó fastruktúrába rendezi az útválasztókat, amelynek ő maga a gyökere. Ennek a hálózati fastruktúrának a neve „legrövidebb út gráf" (SPT, *Shortest Path Tree*). A hálózat útvonalai megfeleltethetőek ezen gráf éleinek. Az útválasztó kiszámítja az egyes útvonalak költségét. A protokoll nagy előnye, hogy a költség mérésébe belefoglalható a lépésszámon kívül sok más információ is, például a kapcsolat sebessége vagy az összeköttetés megbízhatósága.

Osztálymentes (classless) útválasztás

Ahogy a 4. és 5. órán megtanultuk, a TCP/IP útválasztási rendszer alapja a hálózati azonosító fogalma, amely az IP-cím (A, B vagy C) **címosztályaira** épül. Azt is tanultuk az 5. órán, hogy a címosztály rendszernek vannak korlátai, és időnként alkalmatlannak bizonyul arra, hogy egy szolgáltatóhoz hozzá tudjuk rendelni a megfelelő címtartományt. A CIDR (*Classless Internet Domain Routing*) jó alternatívát jelent a címhozzárendelésekhez és útvonal-meghatározásokhoz (lásd az 5. óra CIDR (*Classless Internet Domain Routing*) című szakaszát). A CIDR rendszerben cím/maszk párral tudunk megadni egy gépet, például `204.21.128.0/17`. A maszk adja meg a hálózat-azonosítóhoz tartozó bitek számát.

A CIDR rendszer hatékonyabb útválasztást tesz lehetővé (feltéve, ha az útválasztó protokoll ezt támogatja). A CIDR lecsökkenti az útválasztók között cserélendő információk mennyiségét, mert lehetővé teszi, hogy több osztályhoz tartozó hálózati részt is egyetlen egységként kezeljenek. Az újabb protokollok, mint az OSPF és a BGP4, már támogatják ezt az osztálymentes címzésmódot. Az eredeti RIP még nem, de a RIP II már szintén használható a CIDR-rel.

A verem magasabb rétegei

Az első útválasztók megjelenése óta sokat fejlődtek a hardverek, és természetesen velük együtt a programok is. A hardvergyártók néhány éve felfedezték az IP továbbítás és szűrés magasabb protokollrétegbeli megvalósításának lehetőségeit, előnyeit.

Ahogy a 2-től 7-ig terjedő órákon tanultuk, a verem minden rétege különböző szolgáltatásokat kínál, és különféle információt tárol el a hozzá tartozó fejlécben. A magasabb rétegeket is elérő útválasztónak több információ áll rendelkezésére ahhoz, hogy döntést

hozzon. A szállítási réteget is elérő útválasztó (a forrás és cél kapu-adatára vonatkozó tudása alapján) következtethet az adatok természetére. Az az útválasztó, amely még az alkalmazási réteget is látja, még teljesebb körű ismeretekkel rendelkezhet: feltérképezheti azt az alkalmazást, amely az adatokat küldte, és a protokollt, amelyet ehhez használ.

A magasabb rétegeket is elérő útválasztóknak vannak előnyei. A biztonságot érintő előnyökről még szó lesz a 10. órán („Tűzfalak"). Ennek a technológiának egy másik fontos alkalmazása a szolgáltatás-minőség (QoS, *Quality of Service*) fogalmával kapcsolatos. Vannak olyan fajta adatok, mint például egy internet-telefon ügyféltől származó csomag, amely sokkal érzékenyebb az áthaladási időre, mint más adatfajták (például egy email üzenet). Ha felépült a kapcsolat, akkor az adatcsomagoknak egy ésszerű időkereten belül meg kell érkezniük, máskülönben a telefonvonal csak szaggatottan hallatszik. Az olyan útválasztó, amely hozzáfér az alkalmazási réteghez, elsőbbséget adhat az ilyen csomagoknak a szolgáltatás-minőségi kritériumok alapján.

Ahogy azt a 13. órán majd megtanuljuk („IPv6 – a következő nemzedék") az új IPv6 internet protokoll rendszer más módszereket is rendelkezésünkre bocsát a szolgáltatás-minőség biztosításához. Ennek az órának a megértéséhez azonban most elég annyit tudnunk, hogy sok kifinomult útválasztó van, amely nem csak IP továbbításra képes, hanem számos olyan szolgáltatást nyújt, amely a protokollverem felsőbb rétegeinek információira épül.

Ezeket az útválasztókat általában az OSI modell szóhasználatával (rétegszámozásával) jellemzik. Ahogy megtanultuk a 2. órán („Hogyan működik a TCP/IP"), az OSI modell hét réteget tartalmaz. Egy hagyományos útválasztó a(z alulról számított) harmadik rétegben operál, itt oldja meg az IP adatcsomagok klasszikus továbbítási feladatát. Az ilyen útválasztókat L3 (*Layer 3*, 3. rétegbeli) útválasztóknak hívják. Egy L4 útválasztó a szállítási réteget is eléri. Egy L7 útválasztó az OSI verem legmagasabb rétegét is látja, és így ez képes a kapcsolatban részt vevő alkalmazások támogatására a legtöbb információt felhasználni.

Összefoglalás

Ezen az órán az útválasztás részleteiről volt szó. Tanultunk a távolságvektor alapú és a kapcsolatállapot alapú útválasztásról. Szó volt az IP továbbításról, gerinc útválasztókról, belső és külső útválasztókról. Óránk végén vázolnunk néhány gyakoribb protokollt, melyet a belső útválasztók használnak (a RIP-et és az OSPF-et), valamint megismerkedtünk a magasabb protokollrétegeket használó útválasztás fogalmával.

Kérdések és válaszok

K *Miért kell egy számítógépet felkészíteni az IP továbbításra akkor, ha útválasztóként szeretnénk használni?*

V Az útválasztók olyan adatcsomagokat (is) kapnak, amelyeknek a címe nem egyezik meg a sajátjukkal. Egy átlagos TCP/IP program általában eldobja azokat az adatcsomagokat, amelyek más gépnek vannak címezve. Az IP továbbítás teszi lehetővé azon adatcsomagok elfogadását és feldolgozását is, amelyeknek más alhálózatba kell eljutniuk.

K *Nagyobb hálózatok esetén miért jobb a kapcsolatállapot alapú útválasztás?*

V A távolságvektor alapú útválasztás hatékonysága rohamosan romlik, ha nagy számú útválasztó van a hálózatban. Minden útválasztónak karban kell tartania egy olyan táblázatot, amely az összes célpont adatait tartalmazza. A hálózatra vonatkozó adatok a továbbadás minden egyes lépésében módosulnak. A teljes útválasztó táblát el kell küldeni minden frissítéskor, holott az adatoknak csak egy része érdekes a címzett útválasztó számára.

K *Mi a feladata a külső útválasztónak?*

V A külső útválasztónak az a dolga, hogy útválasztási információkat cseréljen (a saját autonóm rendszerére vonatkozóan) más autonóm rendszerekkel. Amiatt szokták ezt a feladatot kiválasztott útválasztókra bízni, hogy a hálózat többi útválasztójának ne kelljen foglalkoznia más hálózatokra vonatkozó forgalomirányítási kérdésekkel.

K *Miért állítja be a RIP a lépésszám maximumát 15-re?*

V Ha az útválasztók száma túl nagy, akkor probléma származhat a(z egyensúlyi állapothoz képest) túlságosan lassú táblapublikálásból.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **Autonóm rendszer** (*Autonomous system*) – Olyan hálózat, amely egy nagyobb hálózaton belül önálló jellemzőkkel bír, és saját fenntartója van.
- **Külső útválasztó** (*Exterior router*) – Olyan útválasztó, amely egy autonóm rendszerhez tartozik, és útválasztási információkat cserél más autonóm rendszerekkel.
- **Belső útválasztó** (*Interior router*) – Olyan útválasztó, amely szintén egy autonóm rendszerhez tartozik, de a saját autonóm rendszeréhez tartozó más számítógépekkel cserél útválasztási információkat.
- **IP továbbítás** (*IP forwarding*) – Az a folyamat, amelynek során egy IP adatcsomag átkerül egy hálózati felületről ugyanannak a gépnek egy másik hálózati felületére.

- OSPF (*Open Shortest Path First*) – Egy kapcsolatállapot alapú útválasztó protokoll.
- RIP (*Routing Information Protocol*) – Egy távolságvektor alapú útválasztó protokoll.
- Útválasztó protokoll (*Routing protocol*) – Egy vagy több protokoll, amelyet az útválasztók arra használnak, hogy útválasztási információkat gyűjtsenek egymástól.
- SPT (*Shortest Path Tree*) – A hálózatnak egy olyan fagráfra emlékeztető térképe, amit egy OSPF útválasztó állított össze.

9. ÓRA

Kapcsolódás a hálózathoz

Ebben az órában a következőkről lesz szó:

- Telefonos hálózatok
- Szélessávú technológiák, kábeles és DSL kapcsolatok
- WAN hálózatok
- Vezeték nélküli hálózatok
- A kapcsolódáshoz szükséges eszközök

Amint azt az előző órák során megtanultuk, a hálózathozzáférési réteg tulajdonképpen egy a fizikai hálózathoz vezető interfész. No de hogyan is néz ki pontosan ez a bizonyos fizikai hálózat? Mert miután tisztáztuk az összes koncepcionális részletet, biteket, bájtokat, kapukat, protokollrétegeket meg a többit, előbb vagy utóbb elérkezünk arra a pontra, ahol már valami fizikai dolgot kell tapintanunk. Kell valamilyen eszköz, ami a számítógépünket ténylegesen összeköti a helyi hálózattal, és amelyen keresztül elérjük az internetet. Ebben az órában azok közül az eszközök közül fogunk megvizsgálni néhányat, amelyek segítségével TCP/IP hálózatokhoz csatlakozhatunk.

Az óra végére a következőkkel leszünk tisztában:

- Hogyan kommunikálnak a számítógépek a telefonhálózaton keresztül (*dial-up networking*).
- Hogyan működik a kábeltelevíziós szélessávú internetkapcsolat.
- Milyen szolgáltatásokat nyújt egy DSL kapcsolat.

Az óra anyagában röviden áttekintjük azokat az eszközöket is, amelyek segítségével egy TCP/IP hálózat egységes, működő egésszé kapcsolható össze. Szó lesz például a kapcsolókról (*switch*), hubokról és a hidakról (*bridge*).

Telefonos hálózati kapcsolatok

Egészen a közelmúltig a TCP/IP hálózatokhoz vagy az internethez való kapcsolódás legismertebb módja a telefonvonalak használata volt. Az elmúlt néhány évben aztán fokozatosan előretörtek a szélessávú technológiák, megjelentek a kábelmodemek és a különféle DSL kapcsolatok és fokozatosan háttérbe szorították a telefonos hálózatokat. Ezzel együtt a telefonos modem sok területen még mindig a kapcsolódás elsődleges módjának számít, és a számítógépek többsége is támogatja a telefonos kapcsolat kialakítását.

A modem olyan eszköz, amely hálózati kapcsolat létrehozását teszi lehetővé telefonvonalon keresztül. Maga az elnevezés a modulate/demodulate angol szavak rövidítése. A mérnökök eredetileg azért alkották meg ezt az eszközt, mert óriási lehetőséget láttak abban, ha a számítógépek képesek egymással kommunikálni a világ legelterjedtebb kommunikációs hálózatán, a telefonhálózaton keresztül. Az idők során maguk a telefonvonalak is átmentek némi fejlődésen. Manapság egyes vonalakon már lehet digitális adatokat is továbbítani, de azért jócskán akadnak még hagyományos analóg vonalak is. Akár digitális, akár analóg vonalat akarunk is használni, az mindenképpen közös bennük, hogy egyik sincs fölkészítve egy olyan hálózati protokoll használatára, mint amilyen a TCP/IP. A modem feladata az, hogy egy hálózati kapcsolatban a küldő számítógép digitális jeleiből a telefonvonalon továbbítható analóg jeleket állítson elő, a fogadó oldalán pedig végezze el ugyanezt a műveletet visszafelé, ismét a számítógép számára érthető digitális jeleket alkotva a bejövő analóg jelből.

Pont-pont kapcsolatok

Amint arról a 3. órában szó volt, az olyan hálózati technológiák, mint például az Ethernet egészen kifinomult módszereket alkalmaznak arra, hogy lehetővé tegyék a hálózat összes gépe számára a kommunikációs közeg használatát. Egy telefonos kapcsolat esetén ezzel szemben ilyen probléma eleve föl se merül, hiszen a vonalat csak a két,

egymással kommunikáló számítógép használja. Nekik tehát kizárólag egymással kell megosztozniuk a kommunikációs közegen, senki mással. Az ilyen típusú kapcsolatokat pont-pont kapcsolatoknak (*point-to-point connection*) nevezzük (lásd a 9.1. ábrát).

9.1. ábra
Egy pont-pont kapcsolat.

Egy pont-pont kapcsolat a LAN-alapú kommunikációnál annyiban tehát mindenképpen egyszerűbb, hogy nem kell módszert adnia arra, miként osztozzanak meg a számítógépek a kommunikációs csatornán. Ugyanakkor egy telefonvonalon keresztül megvalósított kapcsolatnak mindenképpen vannak bizonyos korlátai. Ezek közül talán a legsúlyosabb az, hogy itt sokkal kisebb átviteli sebesség érhető el, mint például egy Ethernet hálózatban. Ebből egyenesen következik, hogy a telefonos kommunikációt leginkább egy olyan protokoll segítségével lehet megvalósítani, amely a lehető legkevesebb extra információ átvitelét kívánja meg. Minél kevesebb az adminisztráció, annál hatékonyabb lesz a rendszer. Amint arról a későbbiekben részletesen is lesz szó, a modemek fejlődésével nemcsak a sebesség nőtt folyamatosan, hanem a modemes protokollok is egyre újabb feladatokat voltak képesek ellátni.

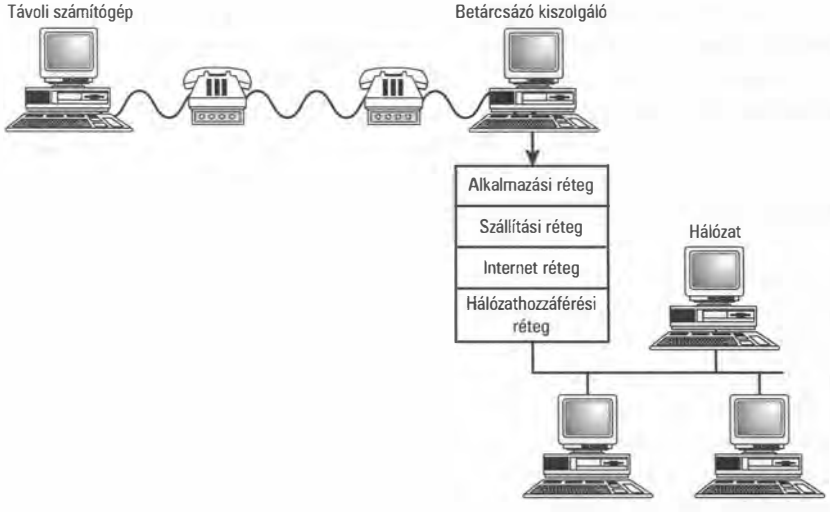
A telefonos kapcsolatokra alkalmas protokollokkal szembeni másik nagy kihívást azoknak a hardver- és szoftverkonfigurációknak a sokfélesége jelenti, amelyeket ki kell szolgálniuk. Egy helyi hálózaton a hálógazdának általában az összes számítógépről meglehetősen pontos ismeretei vannak, illetve ha kell, megfelelően be is állíthatja azokat. Az egész protokollrendszer működése nagyban támaszkodhat a hálózati hardverelemek azonosságára. Egy telefonos kapcsolat ezzel szemben gyakorlatilag a világ bármely pontjáról érkezhet, így a telefonos protokolloknak működniük kell a legkülönbözőbb hardverelemekkel, gépekkel és beállításokkal is.

Modemes protokollok

Az olvasó most bizonyára azon töpreng, hogy a telefonos kapcsolatokban résztvevő két számítógép esetében ugyan miért kell foglalkoznunk mindazzal a rengeteg komplikációval, amit egy TCP/IP verem működtetése jelent. Nos, erre a kérdésre az egyszerű válasz az, hogy elvileg nem kell vele foglalkoznunk.

A korai modemes protokollok csupán egy gépek közti adatátviteli módszert biztosítottak, semmi többet. Ekkor még szó sem volt logikai címekről, hálózatok közti kommunikációról, hibajavításról és mindarról, amit a mai hálózatokban a TCP/IP művel. Erre az egészre

nem hogy szükség nem volt, egyenesen akadályozta volna a telefonos kapcsolatok működését. Később aztán, amikor megjelentek az első helyi hálózatok és az internet, a mérnökök elkezdtek azon gondolkodni, miként használhatnák föl a telefonos kapcsolatokat arra, hogy a felhasználóknak hozzáférést engedjenek a nagyobb hálózatokhoz. Az ilyen távoli hálózati kapcsolatok első megvalósítása tulajdonképpen nem volt több, mint a korai modemes protokollok kiegészítése. Az első telefonos kapcsolódási séma lényege ugyanis annyiban merült ki, hogy a teljes hálózati kommunikáció megszervezéséért az a hálózathoz közvetlenül csatlakozó gép volt a felelős, amelyikhez a telefonvonalon át a másik gép hozzákapcsolódott. Neki kellett az adatokat megfelelően előkészítenie ahhoz, hogy azokat át lehessen küldeni a hálózaton. Explicit, vagy implicit módon, de a telefonvonal túlsó végén levő gép tulajdonképpen nem volt több, mint egy terminál (lásd a 9.2. ábrát), amelyen keresztül a hálózathoz közvetlenül csatlakozó gépet lehetett mindenféle műveletek végrehajtására utasítani. Ez olyannyira igaz volt, hogy az adatoknak a modemes vonalon keresztül történő küldése vagy fogadása eleve egy teljesen más folyamat volt.

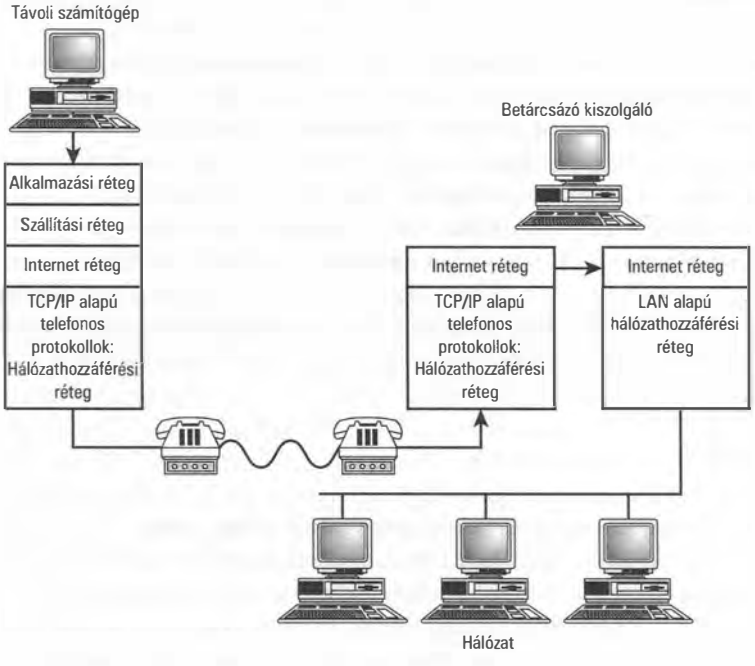


9.2. ábra

A telefonos kapcsolatok egy korai megvalósításának vázlata.

Ezeknek a korai telefonos hálózati sémáknak természetesen mindenféle korlátai voltak. Először is ezek alapvetően egy korábbi, alapvetően centralizált számítási modellt tükröztek, amelyben túlságosan komoly szerep és nagy terhelés hárult a telefonos kapcsolatot biztosító gépre. (Képzeljük el például azt a helyzetet, amikor a 9.2. ábrán bemutatott rendszerben egyszerre több távoli gép kapcsolódik a telefonos kiszolgálóra.) Ráadásul ezek a rendszerek igen kis hatékonysággal használták a távoli gép erőforrásait..

a TCP/IP és az egyéb routolható protokollok megjelenésével a tervezők elkezdtek egy olyan megoldáson gondolkodni, amelyben a távoli számítógép csak a hálózati műveletekért felelős, a betárcsázó kiszolgáló (*dial-up server*) pedig gyakorlatilag egy útválasztóként működik. Ez a megoldás (lásd a 9.3. ábrát) már sokkal közelebb volt a hálózatok új, decentralizált modelljéhez, illetve sokkal inkább összhangban volt a TCP/IP alaptermészetével is. Ebben az elrendezésben a távoli számítógép saját protokollveremmel rendelkezik, a modemes protokollok pedig a hálózathozzáférési rétegben működnek. A betárcsázó kiszolgáló fogadja az ügyfelek adatait, és továbbítja azokat a nagyobb hálózat felé.



9.3. ábra

Egy TCP/IP protokollon alapuló telefonos kapcsolat vázlata.

A fent vázolt fokozatos fejlődés eredményeként végül a telefonos kapcsolati protokollok is elkezdték közvetlenül használni a TCP/IP szolgáltatásait, s így maguk is a hálózati verem részévé váltak. A TCP/IP rendszer két legfontosabb modemes protokollja a SLIP és a PPP.

- **SLIP (Serial Line Internet Protocol)** – Ez egy meglehetősen régi TCP/IP alapú modemes protokoll. Ennek megfelelően ma már van néhány komoly hátránya.
- **PPP (Point-to-Point Protocol)** – Modemes kapcsolatok kiépítéséhez ma ezt a protokollt használják a leggyakrabban. A PPP kezdetben tulajdonképpen a SLIP protokoll egy kifinomultabb változata volt. Számos olyan fontos szolgáltatása van, amivel elődje nem rendelkezett.

Mára a PPP gyakorlatilag teljesen leváltotta a SLIP-et a modemes kapcsolatok területén. Ennek megfelelően a következő szakaszokban ezt a protokollt fogjuk részletesebben megvizsgálni.

A SLIP és a PPP egyaránt olyan alacsonyabb szintű protokollokra épül, amelyek a vonalon haladó jelek modulációjával és demodulálásával foglalkoznak. Ezek a soros vonali kommunikációs protokollok tulajdonképpen olyan szolgáltatásokat nyújtanak, amelyek az OSI modell fizikai rétegének feleltethetők meg.

PPP (Point-to-Point Protocol)

Amikor a szakértők elkezdték megtervezni a PPP szolgáltatásait és működését, már sokkal pontosabb elképzeléseik voltak arról, milyen is lesz majd egyszer az internet, és milyen igényeket támaszt az efféle protokollokkal szemben. Azzal is tisztában voltak, hogy a telefonvonalak idővel egyre jobb minőségűek lesznek, így rajtuk keresztül egyre gyorsabb kommunikációt lehet majd megvalósítani. Éppen ezért bátran terveztek az új protokollba olyan szolgáltatásokat, amelyekhez extra sávszélességre volt szükség. A PPP-vel összességében a SLIP bizonyos hiányosságait szerették volna kiküszöbölni. Ennek részeként a tervezők azt is szerették volna elérni, hogy a PPP képes legyen automatikusan beállítani a két kommunikáló végpontot a kapcsolat közvetlenül felépülésének kezdetén, illetve hogy képes legyen folyamatosan kezelni magát a kommunikációs csatornát.

Hogyan működik a PPP?

A PPP tulajdonképpen egymással együttműködő protokollok olyan gyűjteménye, amelynek célja a modemen alapuló hálózati szolgáltatások kiváltása volt. Tervezése során több RFC is született, a jelenleg használatos változat teljes hivatalos leírása azonban az RFC 1661-es dokumentumban található meg. Néhány további dokumentum a PPP bizonyos összetevőnek működését tisztázta illetve egészítette ki. Az RFC 1661 a PPP összetevőit három átfogó kategóriába sorolja:

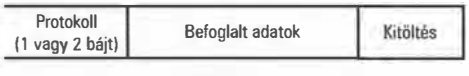
- Módszer többféle protokolltól származó datagramok befoglalására (encapsulation) – A SLIP és a PPP egyaránt datagramokat fogad, amelyeket felkészítenek az internet át történő továbbításra. A SLIP-től eltérően azonban a PPP-nek arra is fel kell készülnie, hogy több különféle protokollrendszertől kaphat datagramokat.
- A kapcsolat létrehozásáért, beállításáért és vizsgálatáért felelős LCP (Link Control Protocol) – A PPP a kapcsolat felépítése során egyezteti a két fél között a szükséges beállításokat, így segítségével elkerülhetők azok a kompatibilitási problémák, amelyek a SLIP működését néha lehetetlenné tették.
- A magasabb szintekhez tartozó protokollrendszerek támogatására hivatott NCP (*Network Control Protocols*) protokollcsalád – A PPP magában foglalhat olyan elkülönült alrétegeket, amelyek egyedi felületeket biztosítanak a különféle protokollcsomagoknak, például a TCP/IP-nek vagy az IPX/SPX-nek.

PPP adatok

A PPP (és a SLIP elsődleges célja a datagramok továbbítása. A PPP számára az egyik nagy kihívás az, hogy több különféle datagramtípust is kezelnie kell. Más szóval a PPP által továbbított adat lehet egy IP datagram, vagy bármilyen más OSI hálózati rétegből származó adatcsomag.

A PPP-t leíró RFC-k a csomag (*packet*) kifejezést használják a PPP által továbbított adatkeretekben (*frame*) továbbított adatok megnevezésére. A csomag tartalma lehet egy IP datagram, vagy bármely más, felsőbb réteg datagramja, illetve állhat olyan adatokból is, amelyek egy tetszőleges, a PPP-n keresztül működő protokoll igényeinek megfelelően vannak formázva. A hálózati technológiákkal kapcsolatos szakirodalomban nagyon gyakran pongyola módon használják a csomag (packet) elnevezést a legkülönfélébb, a hálózaton keresztül továbbított dolgok leírására. Ebben a könyvben igyekeztünk elkerülni ezt a hibát például úgy, hogy a megfelelő helyeken a datagram kifejezést alkalmaztuk. Ugyanakkor a PPP esetében nem minden adatcsomag (*data package*) nevezhető datagramnak, így a vonatkozó RFC-k szóhasználatát átvéve a továbbiakban mi is mindent, a PPP-n keresztül továbbított dolgot csomagnak (*packet*) fogunk nevezni.

A PPP-nek a továbbítani kívánt adatok kezelésén kívül természetesen képesnek kell lennie a saját működését kiszolgáló protokollok adatainak továbbítására is. Egy PPP kapcsolat működése során maguk a kommunikációs eszközös is számos információt kicserélnek egymással. A számítógépeknek LCP csomagokat kell cserélniük ahhoz, hogy felépítsék, kezeljék és lebontsák a köztük fennálló logikai kapcsolatot; a kommunikációs vonalon időnként hitelesítési csomagoknak kell áthaladniuk; maga a PPP NCP csomagok segítségével tartja a kapcsolatot más protokollcsomagokkal. A kapcsolat felépítése során kicserélt LCP csomagok tartalmazzák mindazokat a konfigurációs paramétereket, amelyek valamennyi protokollra vonatkoznak. Ezt követően az NCP protokollok állítják be a PPP kapcsolaton keresztül működő egyes protokollcsomagokra specifikus paramétereket.



9.4. ábra
A PPP adatformátuma

Egy PPP adatkeret formátumát sematikusan a 9.4. ábra mutatja. A csomagban található mezők a következők:

- **Protokoll** – Ez egy egy vagy két bájtos mező, amely a PPP csomagba foglalt másik protokoll numerikus azonosítóját tartalmazza. A befoglalt protokoll lehet az LCP vagy az NCP, szállítható ilyen módon IP csomag, vagy bármely más, az OSI hálózati rétegéhez tartozó protokoll adatai. A különböző protokollok numerikus azonosítóinak listáját az ICANN állítja össze és tartja karban.

- A keretbe foglalt adatok (nulla vagy több bájt) – Az az a konkrét vezérlőcsomag, vagy egy magasabb szintű protokoll csomagja, amit a kérdéses PPP fejléccel ellátva a rendszer szállít.
- Kitöltés (*padding*; opcionális és változó hosszúságú) – További bájtok, amelyek száma a szállított protokolltól függ. Minden szállított protokollnak magának kell meghatároznia, hogy milyen kitöltést kíván alkalmazni a PPP-vel történő szállítás folyamán.

Ha a befoglalt adat olyan protokollhoz tartozik, amely nem része a TCP/IP protokollcsomagnak, akkor annak a tárgyalását nem fogjuk megtalálni ebben a könyvben.

PPP kapcsolatok

Egy PPP kapcsolat életciklusa a következőképpen fest:

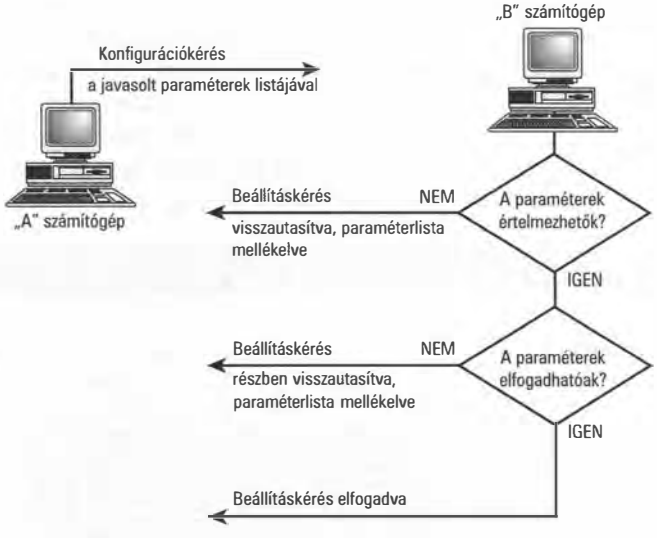
1. Az LCP protokollon lefolytatott egyeztetés során a kapcsolat felépül.
2. Ha az első lépésbe lefolytatott adategyeztetés során kicserélt konfigurációs adatok hitelesítést írnak elő, akkor a két immár kapcsolódott gép a hitelesítési szakaszba lép. Az RFC 1661 erre két különféle eljárást ad meg, a PAP (Password Authentication Protocol) és a CHAP (Challange Handshake Authentication Protocol) protokollokat. Ugyanakkor e kettőn kívül léteznek más olyan hitelesítési protokollok is, amelyeket a PPP támogatni képes.
3. A következő lépésben a PPP NCP csomagok segítségével beállítja azoknak a magasabb szintű (például TCP/IP vagy IPX/SPX) protokolloknak a paramétereit, amelyek az adott vonalon fognak kommunikálni.
4. A PPP továbbítja a felsőbb szintű protokolloktól kapott csomagokat. Ha az első lépésben kicserélt beállítások előírták a vonal minőségének monitorozását, akkor a megfigyelő protokollok is kicserélik az ehhez szükséges adatokat. Az egyes protokollokkal kapcsolatos információkat az NCP képes kicserélni a felek között.
5. A PPP bontja a kapcsolatot a megfelelő terminálo LCP csomagok kicserélésével.

Az LCP (Link Control Protocol)

A PPP hatékonyságát és alkalmazkodóképességét nem kis részben azoknak az LCP műveleteknek köszönheti, amelyek felépítenek, kezelnek és bontanak egy-egy kapcsolatot. Az RFC 1661 alapvetően háromféle LCP csomagot különböztet meg:

- Kapcsolatbeállító csomagok (*Link configuration packets*)
- Kapcsolatbontó csomagok (*Link termination packets*)
- Kapcsolatot karbantartó csomagok (*Link maintenance packets*)

A PPP számos olyan képessége, amellyel a SLIP még nem rendelkezett az LCP-nek köszönhető. A 9.5. ábrán bemutatjuk, miként teszik lehetővé az LCP csomagok, hogy két számítógép kapcsolatot építsen ki egymással. Az első lépésben az „A" számítógép egy LCP konfigurációkérő (LCP Configuration-Request) csomagot küld a „B" számítógépnek. Ez az indító csomag a kérés tényén kívül tartalmazza azokat a javasolt beállításokat is, amelyekben „A" szeretne megegyezni a másik géppel. Ezek a beállítások tartalmazzák a javasolt értékeket a gépek által fogadható legnagyobb adatmennyiségre (Maximum Receive Unit; MRU), a PPP keretekbe foglalható adatmennyiség maximális hosszát, a hitelesítési protokoll azonosítóját, a kommunikáció minőségét megfigyelő protokoll típusát (ez határozza meg, miként kell meggyőződnie a két félnek az adatok sértetlenségéről a kommunikáció során), a tömörítési protokoll beállításait, valamint egyéb fontos paramétereket.



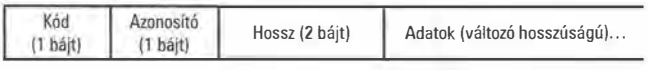
9.5. ábra

A kapcsolat paramétereinek beállítása LCP segítségével

Ha a „B" számítógép az „A" gép által javasolt összes paramétert képes volt értelmezni és azokat elfogadhatónak is találta, akkor ezt a tényt egy Configure-AcK (beállítás visszaigazolva; AcK = acknowledged) csomag visszaküldésével nyugtázza. Ha az „A" gép által küldött valamennyi paraméter értelmezhető ugyan, de ezek közül egyesek „B" számára nem elfogadhatóak, akkor egy Configure-NaK (beállítás részben visszautasítva; NaK = not acknowledged) csomag kerül vissza a kapcsolatot kezdeményezőhöz, valamint azoknak a paramétereknek a listája és „B" által javasolt értéke, amelyekben egyelőre nem történt megegyezés. Ez a folyamat ezután addig folytatódik, amíg a két gépnek a kapcsolat minden paraméterében sikerül megállapodnia.

Az is előfordulhat, hogy az „A" által küldött kérés olyan paramétereket tartalmaz, amelyeket „B" értelmezni sem tud. Ilyenkor a kezdeményezőhöz egy Configure-Rejected (beállítás visszautasítva) csomag kerül vissza azoknak a paramétereknek a listájával együtt, amelyek a megszólított gép számára ismeretlenek.

Egy LCP csomag szerkezetét sematikusan a 9.6. ábra mutatja. Vannak ezen kívül más típusú LCP csomagok is, amelyek a modemes kapcsolat felügyeletéhez használatosak. A 9.6. ábrán a Kód (Code) mező azonosítja az PCP csomag típusát. Az Azonosító (Identifier) mező tartalma egyedileg azonosítja a konkrét LCP csomagot, és abban segít, hogy a rendszer fedésbe tudja hozni a kéréseket a visszaigazolásokkal. A Hossz (Length) mező egyszerűen a csomag hossza. Ezt a továbbított adatok követik, amelyek mennyisége a csomag típusától függ. Az LCP csomagok típuskódjait a 9.1. Táblázatban foglaltuk össze.



9.6. ábra
Az LCP csomagok adatformátuma

9.1. Táblázat *Az LCP csomagok típuskódjai*

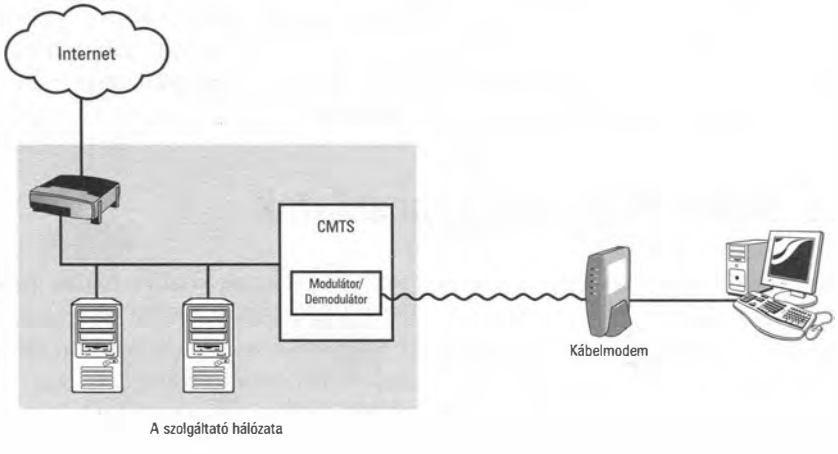
Kód	Leírás
1	Beállítás kérése (Configure-Request)
2	Beállításkérés visszaigazolva (Configure-AcK)
3	Beállításkérés részben visszautasítva (Configure-NaK)
4	Beállításkérés visszautasítva (Configure-Reject)
5	Kapcsolatbontás kérése (Terminate-Request)
6	Kapcsolatbontás visszaigazolva (Terminate-AcK)
7	Kód visszautasítva (Code-Reject)
8	Protokoll visszautasítva (Protocol-Reject)
9	Visszhang kérése (Echo-Request)
10	Visszhang válasz (Echo-Reply)
11	Kérés elvetése (Discard-Request)

Amint azt korábban is említettük, az LCP alapvető feladata a kapcsolatok karbantartása, felépítése és bontása, valamint a beállítások elvégzése. A kapcsolatok bontását a Terminate-Request csomag elküldésével kérheti egy gép, a túloldalnak pedig egy Terminate-AcK típusú csomag visszaküldésével kell ezt visszaigazolnia. A megszólított gép a Code-Reject illetve a Protocol-Reject csomagok segítségével utasíthatja vissza a számára ismeretlen paramétereket illetve protokollokat. Az Echo-Request, Echo-Reply és a Discard-Request csomagok a karbantartás, minőségbiztosítás és hibakeresés során használatosak.

Kábelen közvetített szélessávú kapcsolatok

Az internet nyújtotta szolgáltatások iránti egyre nagyobb fogyasztói igény, illetve a számítógépes rendszerek teljesítményének szakadatlan növekedése arra késztette az ipar képviselőit, hogy a szokványos megoldásokon túl olyan alternatív módszereket kezdjenek keresni a fogyasztói kapcsolatok kiépítésére, amelyek képesek leváltani a lassú és gyakran meglehetősen bizonytalanul működő modemes kapcsolatokat. Ahelyett azonban, hogy felvállalták volna egy teljesen új hálózat kiépítésének kétségtelenül óriási költségét, a keresgélés inkább arra irányult, miként lehetne a meglevő vezetékes hálózatokat fölhasználni erre a célra.

Az egyik olyan, a lakásokig elvezető kábelrendszer, amely kifejezetten alkalmasnak mutatkozott internetkapcsolatok közvetítésére a kábeltelevíziós hálózat volt. A kábelen közvetített szélessávú kapcsolat ma már egészen elterjedtnek számít a világ egyes részein. Egy ilyen kábelmodemes kapcsolat tipikus felépítését a 9.7 ábrán mutatjuk be. A kábelmodem közvetlenül a kábeltelevíziós hálózat koaxiális kábeléhez csatlakozik. A modemnek általában egyetlen Ethernet csatlakozója van, amin keresztül egy PC-hez, vagy egy útválasztóval kombinált kapcsolóhoz csatlakozik. Utóbbi esetben egy egész helyi hálózat építhető ki az internetkapcsolat mögé.



9.7. ábra

Egy tipikus kábelmodemes kapcsolat

Amint azt korábban már említettük, a modem kifejezés a modulátor/demodulátor kifejezéspár rövidítése. A kábelmodem, akárcsak a telefonos modem semmi egyebet nem tesz, mint a számítógép digitális jeleit olyan analóg jelekké alakítja, amelyek hatékonyan továbbíthatók a kábeltelevíziós hálózaton.

Egy másik, már a szolgáltatónál elhelyezett eszköz, az úgynevezett CMTS (Cable Modem Termination System) fogadja a kábelmodem jeleit, és visszaalakítja azokat digitális jelekké. A szolgáltató egy felsőbb szintű internetszolgáltatótól (Internet Service Provider; ISP) bérel egy bizonyos nagyságú sávszélességet, amit egy útválasztó segítségével szétoszt a kábelmodemeken keresztül kapcsolódó előfizetői között. Így valósul meg végül a kapcsolat az otthoni előfizető és az internet többi része között. A szolgáltató ezen kívül nyújthat a kábeles kapcsolaton át egyéb szolgáltatásokat is. Ilyen például a DHCP, amit hálózatra kapcsolódó felhasználók IP címének dinamikus beállításához használnak.

Bár a kábelmodem két különböző adatátviteli közeg között nyújt interfészt, valójában nem tekinthető útválasztónak. Működését tekintve sokkal közelebb áll a hálózati hidakhoz (bridge), amelyekről az óra egy későbbi szakaszában még lesz szó. A kábelmodem a hálózathozzáférési rétegben működve a fizikai cím (MAC address) alapján szűri a forgalmat. Az utóbbi években egyes cégek elkezdtek olyan otthoni útválasztókat is forgalmazni, amelyek fizikailag össze vannak építve egy kábelmodemmel, így előfordulhat, hogy otthonunkban csak egyetlen „dobozt" látunk, amely a teljes kapcsolatot biztosítja.

A korai kábelmodemek esetében a gyártók még teljesen egyedi kommunikációs protokollokat használtak a kábeles közegen való kommunikáció megvalósítására. Az 1990-es évek második felében aztán számos gyártó összefogott, és elkészítették az úgynevezett DOCSIS (Digital Over Cable Service Interface Specification) szabványt, amely egységesítette a kábelen közvetített szélessávú kapcsolatok megvalósítását. Ma már ha egy kábelmodem és egy CMTS egyaránt megfelel a DOCSIS előírásainak, akkor a kapcsolódásuknak elvileg semmi akadálya nem lehet. Ezzel együtt a legtöbb szolgáltató megköveteli, hogy az általunk használt kábelmodem fizikai címét előzetesen regisztráltassuk nála. Erre az illegális használat meggátolása végett van szükség.

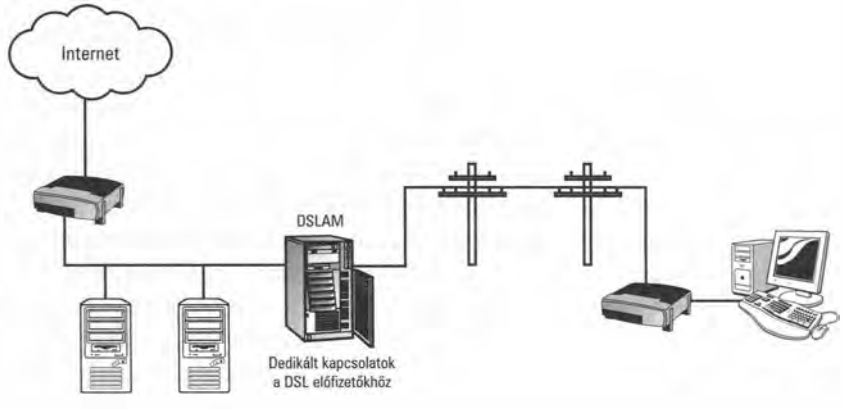
DSL (Digital Subscriber Line) kapcsolatok

Az előfizetők kiszolgálására elvileg alkalmas, már létező hálózatok közül a másik ígéretes jelölt a telefonhálózat volt. A hagyományos telefonos modemek persze már használták ezt a közeget, a telefontársaságok azonban úgy gondolták, hogy egy a fizikai szinten alkalmazott másfajta megközelítéssel nagyobb átviteli teljesítményt tudnak kihozni a létező vonalakból. Így született meg a DSL (Digital Subscriber Line) technológia.

A telefonhálózatok kiépítése során használt csavart érpár valójában sokkal nagyobb sávszélességet biztosít, mint amit a hang analóg továbbításához használnak. A DSL jelátvivő (transceiver) a modemhez hasonlóan olyan eszköz, amely a számítógépek digitális jeleit a telefonvonalon továbbítható analóg jelekké alakítja, ám ezt olyan frekvenciatartományban teszi, amely nem zavarja a hangátvitelt. Ez azt jelenti, hogy a DSL kapcsolat fönntartásához nem kell korlátozni a telefon hagyományos használatát, a szolgáltatás nem foglalja a vonalat, illetve a minőséget rontó interferenciajelenségektől sem kell tartani.

A kábelmodemes kapcsolathoz hasonlóan a DSL esetében is szükség van egy másik, a szolgáltatónál elhelyezett eszközre, amely az analóg jeleket visszaalakítja digitális jelekké, illetve biztosítja a kapcsolatot a szolgáltató hálózatával. Ez az eszköz a DSL esetében a DSLAM (Digital Service Line Access Multiplexer), amely tehát a kapcsolat másik végpontjaként funkcionál (lásd a 9.8. ábrát). A kábeles kapcsolathoz képest lényeges eltérés, hogy míg ott az egy helyi szegmenshez tartozó előfizetők osztoznak a kommunikációs közeg kapacitásán, addig a DSL előfizetőknek dedikált vonaluk van a jelátvivőtől a szolgáltató DSLAM egységéig, ami azt jelenti, hogy a hálózati teljesítmény kevésbé érzékeny az általános forgalomnövekedésre.

A DSL kapcsolatoknak számos típusa létezik. Az otthoni és kis irodai előfizetők körében a legnépszerűbb forma az ADSL (Asynchronous DSL), ám létezik HDSL (High bit-rate DSL), VDSL (Very High bit-rate DSL, SDSL (Sysmmetric DSL; itt a le- és feltöltés sebessége megegyezik) és IDSL (ISDN over DSL) is. A protokollok szempontjából szemlélve a DSL kapcsolat működése nagyban függ az alkalmazott eszközöktől illetve a konkrét megvalósítástól. Egyes DSL eszközöket eleve összeépítenek hálózati kapcsolókkal illetve útválasztókkal. Más eszközök inkább hidakként működnek, és a kábelmodemekhez hasonlóan a hálózathozzáférési rétegben működve a fizikai cím (MAC address) alapján szűrik a forgalmat. Kifejezetten gyakori, hogy a DSL eszközök a továbbított adatokat valamilyen pont-pont kapcsolati protokoll (ilyen például a korábban tárgyalt PPP) kereteibe ágyazva kezelik. Az úgynevezett PPPoE (PPP over Ethernet) protokoll használata például egy egészen gyakori megoldás a DSL eszközök esetében.



9.8. ábra

Internetkapcsolat kiépítése DSL segítségével

WAN (Wide Area Network) hálózatok

Az olyan cégek és szervezetek hálózati igényei, amelyek rengeteg számítógéppel és esetleg több telephellyel is rendelkeznek értelemszerűen nem elégíthetők ki olyan technológiák segítségével, mint a telefonos hálózat, vagy a DSL. Az ilyen típusú előfizetőknél a leglényegesebb kérdés az, hogy miként lehet úgy összekapcsolni a fiókirodákat és telephelyeket úgy, hogy az ehhez használt a vonalak ne csak a szükséges teljesítményt biztosítsák, hanem a helyi hálózatokkal egyenértékű biztonságot is nyújtsanak. Ezek voltak azok a kérdések, amelyek végül elvezettek a nagy kiterjedésű hálózatok (Wide Area Network; WAN) kifejlesztéséhez.

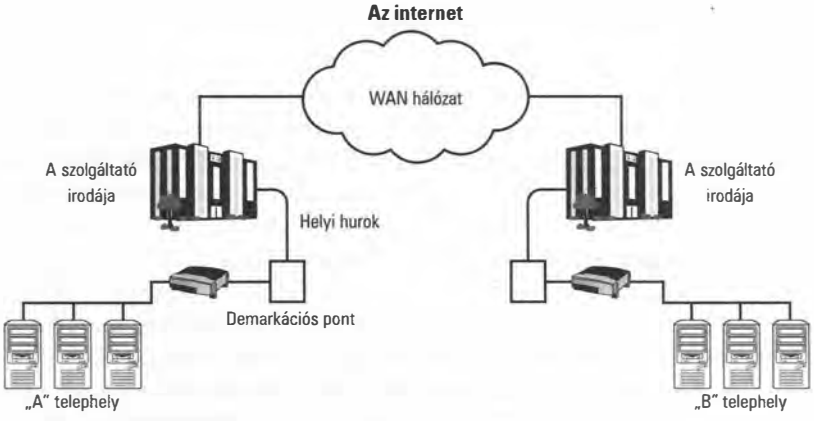
A WAN technológiák gyors, nagy sávszélességű hálózati összeköttetést képesek biztosítani akkor is, ha az összekötni kívánt pontok egymástól viszonylag távol helyezkednek el. Bár a WAN hálózatok teljesítménye nem éri el a LAN-ok nyújtotta sebességét, mindenképpen sokkal megbízhatóbb és biztonságosabb kapcsolatok kiépítését teszik lehetővé, mint ha a hagyományos technológiák segítségével a nyílt internetet használnánk erre a célra. WAN stílusú hálózatok biztosítják a világméretű vállalatok belső kommunikációját, sőt, egyes esetekben tulajdonképpen az internet néven ismert nagy, rejtélyes hálózati konglomerátum kisebb alkotóelemei is WAN technológiára épített nagy sávszélességű részhálózatok.

Íme néhány a ma rendelkezésünkre álló WAN technológiák közül:

- Frame Relay
- ISDN (Integrated Service Digital Network)
- HDLC (High-Level Data Link Protocol)
- ATM (Asynchronous Transfer Mode)

Bár ezeknek a technológiáknak a részletes ismertetése egyenként is megtöltene egy-egy könyvet, végső soron egyik sem más, mint egy-egy másfajta megvalósítása annak a fizikai kommunikációs kapcsolatnak, amelyen át a TCP/IP hálózathozzáférési rétege működhet. A WAN protokollok csaknem kivétel nélkül az OSI modellen alapulnak, tehát ezen a ponton talán érdemes fölidézni, hogy a hálózathozzáférési réteg (Network Access Layer) az OSI fizikai (Physical Layer) és adatkapcsolati (Data Link Layer) rétegének feleltethető meg. (Ez utóbbiakat szokás Layer 1 és Layer 2 néven is említeni.)

Egy tipikus WAN hálózat vázlatos felépítését a 9.9. ábrán láthatjuk. Magát a WAN-t egy olyan szolgáltató üzemelteti, amelynek egyaránt van fizikai kapcsolata az internettel és az előfizető fióktelepivel. A szolgáltató irodáját egy helyi hurok kapcsolja össze az úgynevezett demarkációs ponttal (*demarcation point*), amely nem más, mint a hálózatnak az a helye, ahol maga az előfizető csatlakozik a rendszerhez. A kapcsolódáshoz szükséges speciális berendezéseket, illetve a helyi hálózatot a WAN-nal összekötő útválasztót ebben az esetben az előfizető biztosítja.



9.9. ábra

Egy tipikus WAN hálózat

A demarkációs ponttól kezdődő hálózati szakaszra a szolgáltató egy megadott sávszélességet illetve szolgáltatási minőséget garantál. A szolgáltatással kapcsolatos díjazás módja változó. Létezik dedikált bérelt vonalon át megvalósított WAN kapcsolat, de az is előfordul, hogy a megrendelő csak azért a hálózati teljesítményért fizet, amit ténylegesen használt.

Vezeték nélküli hálózatok

A technológiai fejlődés mára elérte azt a szintet, hogy a felhasználóknak még a kábelezéssel, és az eszközök Ethernet csatlakozókon keresztül történő összekapcsolásával sem föltétlen kell foglalkozniuk. Számos olyan vezeték nélküli kommunikációs szabványt dolgoztak ki, amelyek a TCP/IP protokollcsomaggal is képesek együttműködni.
Az elkövetkező szakaszokban a következő technológiákról lesz szó:

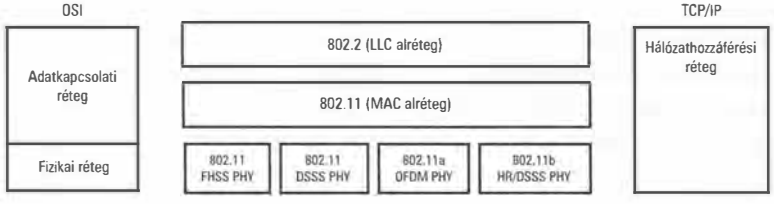
- 802.11 szabványú hálózatok
- WAP
- Mobil IP kommunikáció
- Bluetooth technológia

Hogy ezek a viszonylag új technológiák miként épülnek be a termékekbe és szolgáltatásokba, az nagyban függ a konkrét gyártótól illetve szolgáltatótól. Ennek megfelelően a következőkben csak az általános elvekkel kívánjuk megismertetni az olvasót.

802.11 hálózatok

Amint azt a harmadik órában megtanultuk, a hálózat konkrét fizikai felépítésével kapcsolatos részletek a TCP/IP protokollverem hálózathozzáférési rétegére „tartoznak". A legegyszerűbb szinten tehát egy vezeték nélküli hálózatot úgy képzelhetünk el, mint egy teljesen közönséges hálózatot, amelynek a hálózathozzáférési rétegében történetesen egy rádióhullámokon alapuló kapcsolat működik. A manapság oly népszerű IEEE 802.11 specifikáció nem más, mint egy ilyen rétegnek a modellje.

A 802.11 protokollverem vázlatos felépítését a 9.10. ábra mutatja be. A hálózathozzáférési réteghez tartozó vezeték nélküli komponensek egyenértékűen azon egyéb hálózati architektúrák megfelelő komponenseivel, amelyekről az eddigiek során már volt szó. Ami azt illeti, a 802.11 szabványt sokan hívják vezeték nélküli Ethernetnek is, mivel lényegét tekintve rettentően hasonlít az IEEE 802.3-ban leírt igazi Ethernet szabványra.



9.10. ábra

A 802.11 protokollok a TCP/IP hálózathozzáférési rétegéhez tartoznak

Amint a 9.10. ábrán is látható, maga a 802.11 specifikáció tulajdonképpen teljes egészében az OSI referencia modell MAC alrétegét valósítja meg, amely maga az OSI modell adatkapcsolati rétegének (Data Link Layer) egyik alkotója. Emlékezzünk vissza továbbá, hogy a második órában tisztáztuk: az OSI adatkapcsolati és fizikai rétegei együttesen a TCP/IP hálózathozzáférési rétegének felelnek meg. A fizikai rétegben látható számos különféle opció a különféle vezeték nélküli üzenetszórási megoldásoknak felel meg. (FHSS = Frequency Hopping Spread Spectrum ; DSSS = Direct Sequence Spread Spectrum ; OFDM = Orthogonal Frequency Division Multiplexing ; HR/DSSS = High Rate Direct Sequence Multiplexing).

A legfontosabb olyan tulajdonság, amely a vezeték nélküli hálózatokat megkülönbözteti hagyományos, vezetékes megfelelőiktől az, hogy a hálózati csomópontok itt mozoghatnak. Ez egyben azt is jelenti, hogy a hálózatnak képesnek kell lennie kezelni azt a szituációt, amikor a benne résztvevő eszközök helye megváltozik. Amint arról korábban volt szó, a TCP/IP tervezésénél kezdetben szó sem volt vezeték nélküli hálózatokról, így arról sem, hogy a csomópontok mozoghatnak. Az alapkoncepció kötött helyzetű hálózati csomópontokra épült. Ennek megfelelően ha egy számítógépet áthelyezünk az egyik

hálózati szegmensből egy másikba, akkor vagy megváltoztatjuk az összes lényeges beállítását – beleértve például a címét – is, vagy egyszerűen nem fog működni. Ezzel szemben egy vezeték nélküli hálózatban a résztvevők szinte állandó mozgásban vannak. És bár igaz, hogy a hagyományos Ethernet hálózatokkal kapcsolatos számos alapkoncepció ebben a helyzetben is használható marad, azért az is világos, hogy a vezeték nélküli kommunikáció egy egészen más, bonyolultabb szituáció, amelynek kezeléséhez új stratégiákra van szükség.

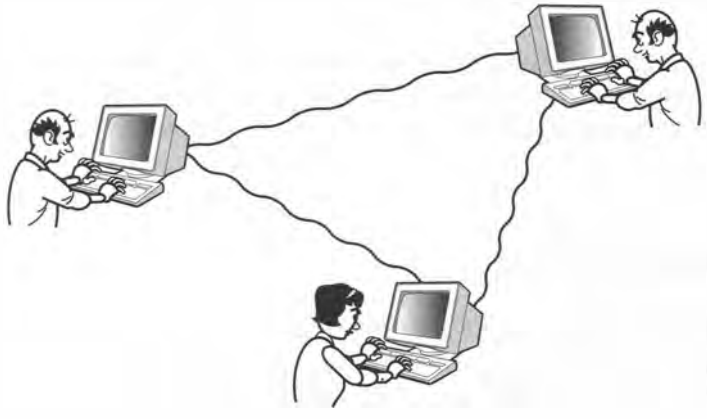
A 802.11 valójában számos különféle szabvány gyűjtőneve. Az eredeti 802.11-es szabvány 1997-ben jelent meg, és legfeljebb 2 Mbps-os átviteli sebességet tett lehetővé a 2,4 GHz-es frekvenciasávban. A 802.11a szabvány már 54 Mbps sebességet tesz lehetővé 5 Ghz-en kommunikálva, míg a 802.11b 5,5 Mbps-os, illetve 11 Mbps-os kapcsolatokat különböztet meg és szintén a 2,4 GHz-es tartományban kommunikál.

Független és infrastrukturális hálózatok

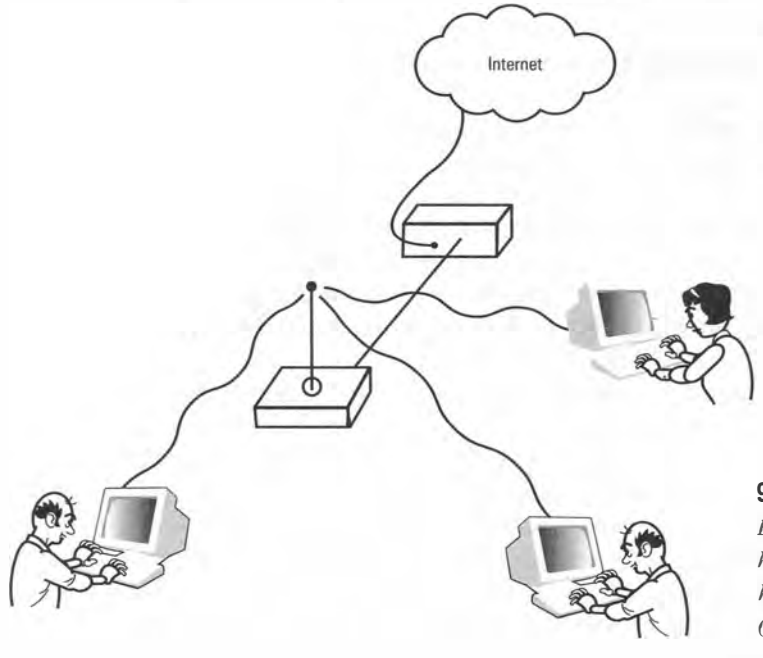
A legegyszerűbb vezeték nélküli hálózat két vagy több olyan számítógépből áll, amelyek vezeték nélküli hálózati adapterei egymással közvetlenül kommunikálnak (lásd a 9.11. ábrát). Az ilyen típusú hálózat, amit angolul BSS-nek vagy IBSS-nek rövidítenek (Independent Basic Service Set) gyakorta elegendő számítógépek olyan kis számú csoportjának, amelyek egymáshoz kellően közel működnek. Az ilyen hálózat tipikus példája az a helyzet, amikor valaki a hordozható gépével a hóna alatt épp hazatér egy üzleti útról, és szeretné az összegyűjtött adatokat átmásolni az asztali gépére vezeték nélküli hálózaton keresztül. Független BSS hálózatok néha spontán a semmiből is bírnak keletkezni, ha például munkatársak találkoznak egy konferencián vagy egy irodában, körbeülnek egy asztalt és a megbeszélés idejére hordozható számítógépeiket is összekapcsolják, hogy adatokat tudjanak cserélni. A független BSS hálózat működését és képességeit tekintve természetesen meglehetősen komoly korlátokkal bír. Hatékonysága értelemszerűen függ a résztvevő gépek közelségétől, ami pedig az adatáramlást és a külső kapcsolatokat illeti, az ilyen hálózatban semmiféle infrastruktúra nem gondoskodik a kapcsolatok kialakításáról, sőt egy LAN-hoz vagy az internethez sem tudnak kapcsolódni a résztvevői.

A vezeték nélküli hálózatok másik formája, az infrastrukturális hálózat (Infrastructure BSS) sokkal gyakoribb a cégek, szervezetek hálózataiban, illetve manapság egyre népszerűbb megoldás a kis otthoni hálózatok, illetve netkávézók kialakítására is. Ez utóbbi alapvetően az olcsó vezeték nélküli hálózati útválasztók tömeges megjelenésének köszönhető. Az infrastrukturális hálózat működése egy kötött helyű eszköztől, a hozzáférési ponttól (access point) függ. Ez az eszköz egyfajta központként kezeli a gépek közti kapcsolatokat (lásd a 9.12. ábrát). A hozzáférési pont a vezeték nélküli hálózat résztvevőivel természetesen rádióhullámok segítségével kommunikál, azonban általában van egy vezetékes kapcsolata is az internet felé is, amit a vezeték nélküli ügyfelek között

megoszt. A hálózat résztvevői a hozzáférési ponton keresztülkommunikálnak. Ez azt jelenti, hogy ha egy vezeték nélküli gép egy másik vezeték nélküli gépnek kíván üzenetet küldeni, akkor ezt előbb a hozzáférési pontnak küldi el, az pedig továbbítja az igazi címzettnek. Ha egy gép a hálózat hagyományos részével kíván üzenetet cserélni, akkor a hozzáférési pont hídként (*bridge*) viselkedik. Összefoglalva tehát a hozzáférési pont minden hagyományos hálózaton levő gépnek küldött adatkeretet továbbít, és minden vezeték nélküli technológiával kommunikáló gépnek szóló keretet megtart.

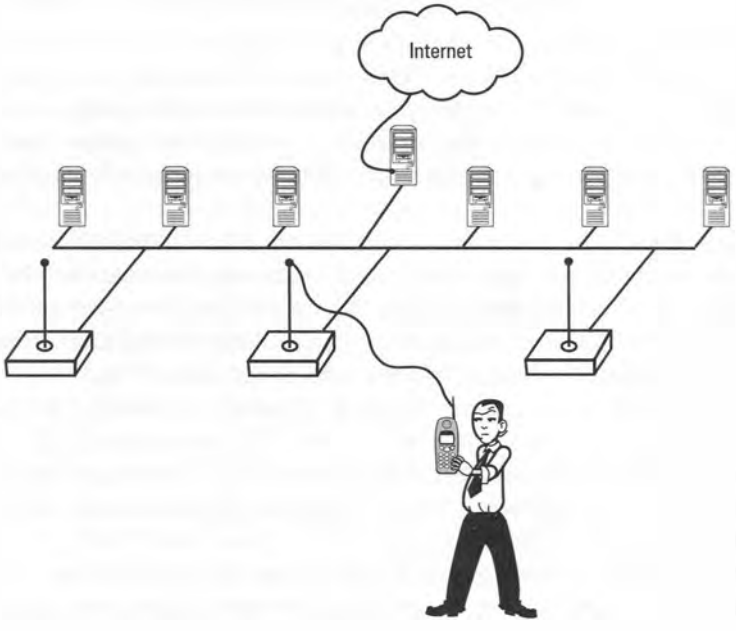


9.11. ábra
Egy független vezeték nélküli hálózat működése



9.12. ábra
Egy infrastrukturális hálózat egy vagy több hozzáférési pontot (access point) tartalmaz.

A 9.12. ábrán látható hálózat számítógépei gyakorlatilag ugyanúgy működhetnek, mintha egy közönséges, vezetékekből felépített hálózat elemei lennének. Az infrastrukturális hálózatok több hozzáférési pontot is tartalmazhatnak. Ennek különösen akkor láthatjuk nagy hasznát, ha egy nagyobb területen elhelyezkedő gépeket akarjuk összekapcsolni egyetlen vezeték nélküli hálózattá. Ilyenkor az egyes hozzáférési pontokat közönséges Ethernet vezetékekkel lehet összekapcsolni, és megfelelően szét kell szórni nőket az adott területen (lásd a 9.13. ábrát).



9.13. ábra
Infrastrukturális vezeték nélküli hálózat több hozzáférési ponttal

A 802.11 szabványt eleve úgy fogalmazták meg, hogy az alkalmas legyen a 9.13. ábrán vázolt hálózatok kiszolgálására is. A dolog lényege, hogy a barangoló eszköz soha ne veszítse el a hálózati kapcsolatot, akárhol is legyen a hálózat lefedettségi területén belül. Ezzel kapcsolatban az első dolog, amit felfedezhetünk az, hogy a dolog működéséhez a hálózatnak folyamatosan tudnia kell, melyik hozzáférési ponton keresztül szólítsa meg az eszközt, ha valaki üzenetet akar neki küldeni. A helyzetet természetesen az is bonyolítja, hogy az eszköz mozoghat, vagyis a megfelelő hozzáférési pont minden különösebb értesítés nélkül megváltozhat. Szintén lényeges észrevétel, hogy a hagyományos forráscím-célcím páros itt már nem elegendő információ az üzenet sikeres kézbesítéséhez. Ennek megfelelően a 802.11 keretekben négyféle cím tárolódik:

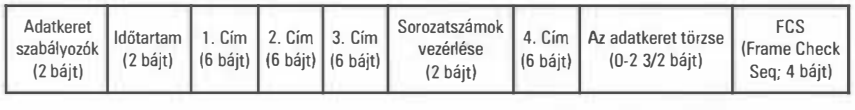
- **Célcím (*Destination address*)** – Annak az eszköznek a címe, amelynek az adatkeret szól.
- **Forráscím (*Source address*)** – Annak az eszköznek a címe, amely az adatkeretet küldi.
- **Fogadó címe (*Receiver address*)** – Annak a vezeték nélküli eszköznek a címe, amelynek fel kell dolgoznia az elküldött adatkeretet. Ha az üzenet egy másik vezeték nélküli eszköznek szól, akkor a fogadó címe ennek az eszköznek a címe lesz. Ha az üzenet

címzettje egy hagyományos hálózat tagja, akkor a fogadó címe a hozzáférési pont címe. Első közelítésben ő kapja meg a adatkeretet, majd továbbítja azt az Ethernet hálózatba.

- Jelátvivő címe (*Transmitter address*) – Annak az eszköznek a címe, amely a vezeték nélküli hálózatba továbbította a kérdéses adatkeretet.

A 802.11 szabványnak megfelelő adatkeretek felépítését a 9.14. ábra mutatja. A lényegesebb mezők jelentése a következő:

- Adatkeret szabályozók (*Frame control*) – Kisebb mezők gyűjteménye, melyek között van a protokoll verziószámát és az adatkeret típusát azonosító érték, illetve egyéb olyan adatok, amelyek a keret tartalmának értelmezéséhez szükségesek.
- Időtartam (*Duration/ID*) – Ennek a mezőnek az értéke az átvitel becsült idejét tartalmazza. Ugyanebben a mezőben lehet elkérni a pufferelt kereteket a hozzáférési ponttól.
- Címmezők (*Address fields*) – 48 bit széles mező, amely a megfelelő fizikai címeket tartalmazza. Ahogy azt korábban említettük, a 802.11 szabvány bizonyos helyzetekre négy különböző cím megadását írja elő. A címmezőket a rendszer a keret típusától függően különbözőképpen használja. Ugyanakkor az első mező általában a fogadó, míg a második jellemzően a jelátvivő címét tartalmazza.
- Sorozatvezérlés (*Sequence control*) – A fragmensszámot és a keret sorozatszámát tartalmazza (előbbit a rendszer a töredezett keretek helyreállításához használja).
- Az adatkeret törzse (*Frame body*) – Az adatkeretben továbbított hasznos adatok. Amint arról a második órában már volt szó, a kerettel továbbított adatok tartalmazzák a magasabb szintű protokollok fejléceit is.
- FCS (*Frame Check Sequence*) – Egy CRC (Cyclic Redundancy Check) érték, amit az átviteli hibák, illetve a szándékos módosítások ellenőrzésére használ a rendszer.



9.14. ábra

Egy 802.11 adatkeret felépítése

Érdemes megjegyezni, hogy mivel a 802.11 egy a hálózathozzáférési réteghez tartozó protokoll, a keretek a harmadik órában említett 48 bites fizikai címeket tartalmazzák, nem az IP címeket. Amint az eszköz mozog a hálózatban, mindig regisztrálja magát a legközelebbi hozzáférési ponton. (Ez műszakilag azt jelenti, hogy az érzékelhető hozzáférési pontok közül azt választja ki, amelyiknek a legerősebb a jele és a legkisebb interferenciát mutatja.) Ezt a regisztrációs folyamatot nevezik asszociációnak (*association*). Ha az eszköz mozgása során egy másik hozzáférési ponthoz kerül közelebb, módosítja a kapcsolatát (*reassociation*). Ez az asszociációs folyamat az, amely lehetővé teszi, hogy a hálózat mindig a megfelelő hozzáférési ponton keresztül szólítsa meg a az eszközöket.

A 802.11 szabványt alkalmazó eszközök egymással való kompatibilitásának fönntartása érdekében megalakult egy WECA (Wireless Ethernet Compatibility Alliance) nevű szervezet, amely hitelesíti a forgalmazott készülékeket. Ahhoz, hogy egy eszköz ilyen Wi-Fi (Wireless Fidelity) minősítést kapjon, meg kell vizsgálni, hogy képes-e kapcsolódni más, már hitelesített eszközökhöz. A WECA-ról és a Wi-Fi tanúsítványról bővebb információt a *http://www.wi-fi.org* címen találhatunk.

A 802.11 szabvány biztonságossá tétele

Bizonyára senkit nem lep meg, hogy a vezeték nélküli hálózatok különösen is sok biztonsági rést tartalmazhatnak. Egy hagyományos hálózat lehallgatásához legalább fizikailag rá kellett csatlakozni az átviteli csatornára. Ezzel szemben a vezeték nélküli hálózat sokkal sebezhetőbb: a közvetítési körzetben bárhonnan elérhető. Nemcsak bele lehet hallgatni a hálózaton átjutó adatokba, hanem a vállalkozó kedvű támadó akár részt is vehet a hálózat eseményeiben, ha nincs megfelelő védelem ez ellen. Ezeknek a problémáknak az orvoslására az IEEE kifejlesztett egy biztonsági protokollt a 802.11-hez.
A Vezetékessel Egyenértékű (Biztonságú) Hálózat (*Wired Equivalent Privacy*, WEP) szabványa mára már kissé elavult. Eredetileg arra tervezték, hogy a hagyományos vezetékes hálózathoz hasonló biztonsági szintet lehessen létrehozni vezeték nélküli hálózatokon. A WEP a következőket célozta meg:

- Megbízhatóság – védelem a lehallgatás ellen
- Integritás – az adatok sérülése elleni védelem
- Hitelesítés – annak biztosítása, hogy a kommunikáló felek valóban azok, akiknek mondják magukat, és hogy a hálózaton történő műveletekhez megvan a szükséges jogosultságuk.

A WEP a megbízhatóságot és integritást az RC4 algoritmusnak nevezett titkosítással éri el. A küldő fél előállít egy integritás-ellenőrző értéket (*Integrity Check Value*, ICV), amelyet az adott keret adattartalmából lehet kiszámítani egy szabványos eljárással. Az ICV-t azután titkosítják az RC4 algoritmussal, és a kapott értéket (a kerettel együtt) elküldik a címzettnek. A címzett visszafejti a keretet és kiszámítja az ICV-t. Ha a számolt érték megegyezik azzal, ami a kerettel együtt érkezett, akkor biztosak lehetünk benne, hogy az adatcsomag változatlanul célba ért.

A WEP sajnos nem minden tekintetben nyerte el a biztonsági szakértők elismerését. Manapság a legtöbb szakértő alacsony hatékonyságúnak tekinti a WEP-et. Néhányuknak az RC4 titkosítás megvalósítása ellen van kifogása. A WEP elméletileg 64 bites kulcsot használ, de ebből 24 bit az inicializációra van lefoglalva. Mindössze 40 bit használható tehát a megosztott kulcsos titkosításhoz. A legtöbb szakértő nem tartja megbízhatónak a 40 bites kulcsot, így a WEP nem elégséges a hatékony védelemhez. A szakértők arra is rámutattak, hogy gondok vannak a kulcskezeléssel és a titkosítás kezdetén használt 24 bites inicializációs vektorral.

A WEP továbbfejlesztéseként alakult ki a WEP2, amelyben az inicializációs vektor 128 bites. Ez már Kerberos hitelesítést használ a titkos kulcsok használatához és szétosztásához. A WEP2 azonban nem oldja meg az összes problémát, amit a WEP felvet. A WEP hibáinak kiküszöbölésére más protokollok is megjelentek (példáiul a Kiterjeszthető Hitelesítési Protokoll; *Extensible Authentication Protocol*; EAP).

A 802.11i szabványtervezet 2004-ben jelent meg, azzal a szándékkal, hogy egy jobb és biztonságosabb vezeték nélküli protokoll alakulhasson ki. Ez 2007-ben került be a 802.11 szabványba. A WPA2-ként ismert új megközelítés az RC4 helyett szabványos AES blokk-titkosítást használ, és biztonságosabb folyamatokkal történik a hitelesítés és a kulcsmegosztás. A WPA2 nagy előrelépésnek tűnik a vezeték nélküli hálózatok biztonsága tekintetében – e sorok írásakor azonban még világszerte sok helyen használnak WEP-et.

Számos vezeték nélküli eszköz többféle biztonsági szintet támogat. Sok vezeték nélküli útválasztó lehetővé teszi a számítógépek fizikai MAC címének a megadását – ekkor csak ezek a gépek dolgozhatnak a hálózaton. Ezek a megoldások gyakran igen hasznosak akkor, ha meg akarjuk akadályozni, hogy a szomszédunk a mi sávszélességünk rovására dolgozzon. Legyünk azonban tisztában azzal, hogy tapasztalt betörők ezt a fajta védelmet ki tudják játszani.

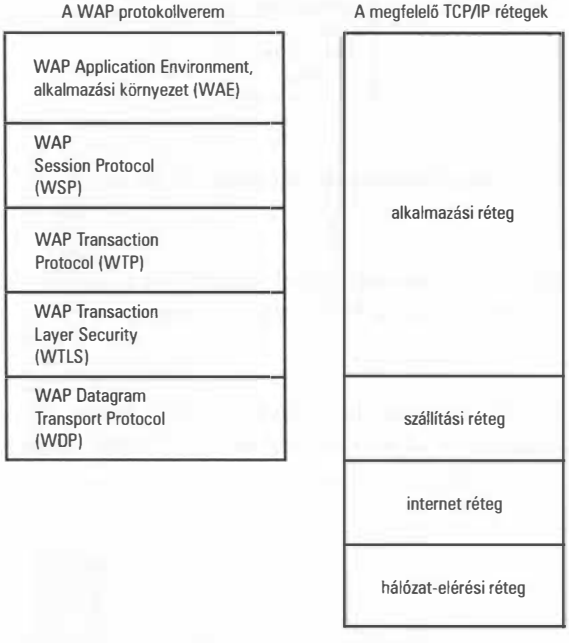
Wireless Application Protocol (WAP)

A 802.11 és az ehhez hasonló szabványok azt a célt állítják a középpontba, hogy a helyi hálózatban a vezeték nélküli eszközöket sikerüljön összehangolni a hagyományos hálózati eszközökkel. Az igazi „mobil internet" elv azonban ennél többet jelent: a felhasználó bárhova mehet, kézi eszközével el tudja érni az internetet a mobil telefonhálózat révén – ennek teljesítése azonban igényel némi többletmunkát. A Wireless Application Protocol („*Vezeték Nélküli Alkalmazási Protokoll*") protokollverem kifejezetten vezeték nélküli eszközökhöz készült. Míg a 802.11 hálózatok alapvető tulajdonságaikat illetően az Ethernet hálózatokra hasonlítanak, a WAP gyökeresen eltér a hagyományos TCP/IP hálózatoktól. A WAP valójában egy magasabb szintű, alkalmazásokat érintő protokollréteg. Arra tervezték, hogy mobil eszközök számára lehessen információt átadni (és tőlük átvenni).

A WAP protokollverem protokolljai a „vezeték nélküli világ" igényeihez alkalmazkodva (és tőlük motiválva) jöttek létre. A WAP specifikáció tartalmaz egy sajátos XML alapú jelölőnyelvet is, melyet Wireless Markup Language-nek (WML) hívnak. Kifejezetten olyan webes tartalmak megjelenítésére hozták létre, amelyeket a hordozható készülékek kis képernyőjén nézhetnek a felhasználók. A WAP saját protokollrétegeket biztosít, amelyek nagyjából az OSI verem felső rétegeinek felelnek meg. A tervezők azonban nem féltek bizonyos mértékig eltávolodni a szigorúan vett OSI szabványtól, hogy a vezeték nélküli

hálózati környezet speciális igényeihez igazodjanak. A WAP protokollrétegek (és az egyes protokollok) a 9.15 ábrán láthatóak. A könyvünkben leírt más protokollokhoz hasonlóan a WAP protokollok is csak egy specifikációt adnak, amelyet a különféle szoftvergyártók különböző módon valósítanak meg. A WAP protokollok a következőket foglalják magukban:

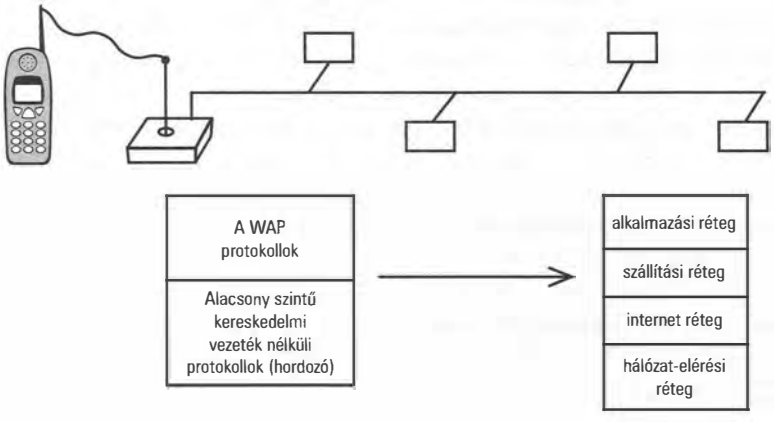
- WAP munkamenet protokoll (WAP Session Protocol, WSP) – A HTTP WAP-os megfelelője. A WSP biztosítja az alkalmazások közti adatcsere rendszerét.
- WAP tranzakciós protokoll (WAP Transaction Protocol, WTP) – Szolgáltatások közti kapcsolatfelvételt („kézfogást") és nyugtázást biztosító protokoll. Ezekkel lehet WAP tranzakciókat kezdeményezni vagy jóváhagyni.
- WAP biztonsági tranzakciós réteg (WAP Transaction Layer Security, WTLS) – SSL alapú biztonsági protokoll (23. óra: „A TCP/IP biztonsága").
- WAP adatcsomag-átviteli protokoll (WAP Datagram Transport Protocol, WDP) – Kapcsolat nélküli adatátviteli réteg protokoll, ami az UDP-hez hasonlít (6. óra: „Az adatátviteli réteg").



9.15 ábra
A WAP protokollverem

A WAP protokollok a protokollverem felsőbb rétegeivel hozhatók összefüggésbe. Figyeljük meg, hogy a WDP hasonló az UDP protokollhoz, amely a TCP/IP szállítási rétegében található. A WAP protokollverem nagy része a TCP/IP alkalmazási rétegében található. Nem egyszerű azonban pontosan megfogalmazni, hogy milyen viszonyban is van egymással a WAP és a TCP/IP.

Mivel a vezeték nélküli hálózatok jelentősen lassabbak és megbízhatatlanabbak a vezetékeseknél, a WAP protokollokat arra helyezték ki, hogy a lehető legnagyobb teljesítményt lehessen kihozni a segítségükkel. Vannak olyan WAP protokollok, amelyek bináris formátumot használnak, s ezeket át kell fordítani szöveges formátumúvá a TCP/IP protokollok számára, amikor a WAP-os eszközök internetes dokumentumokat fogadnak. A **WAP átjáró** az az eszköz, amely a WAP protokollon érkező információt átfordítja interneten is értelmezhető formátumúvá (9.16 ábra).



9.16 ábra

A WAP átjáró fordítja át a WAP protokollon érkező információt interneten is értelmezhető formátumúvá.

A WAP protokollkészlet magába foglal néhány más protokollt és nyelvet is, amelyek nincsenek megjelenítve a 9.15 ábrán; ilyen például a WMLScript szkriptnyelv és a WBMP képformátum.

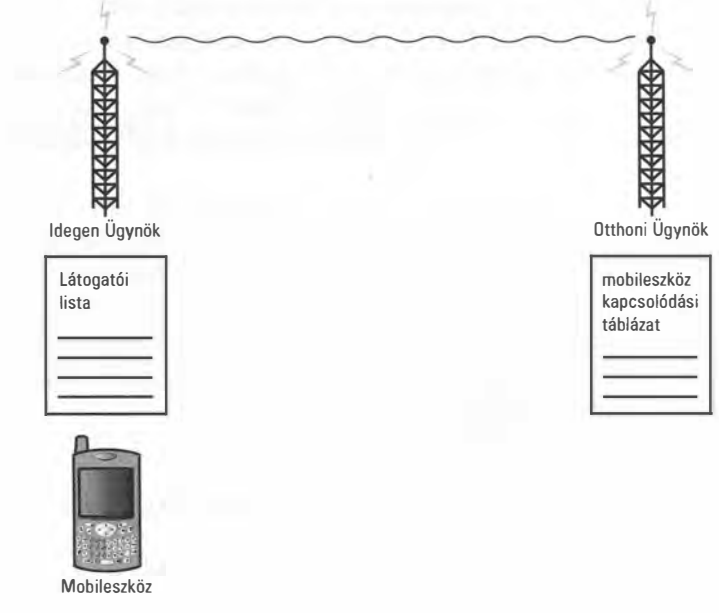
Az újabb WAP szabványok erősebb TCP/IP kompatibilitást javasolnak, valamint (az XHTML-en keresztül) szorosabb igazodást az XML-hez és a HTML-hez. Előbb-utóbb az XHTML le fogja cserélni a WML-t, a WAP jelölőnyelvét.

Mobil IP

Bizonyára belegondolt már az olvasó, hogy a világszerte elterjedőfélben levő mobil eszközök komoly kihívást jelentenek az internetes lekérések teljesítése szempontjából. Az internetcímek rendszere hierarchikus felépítésű, azzal a feltevéssel, hogy a címzett azon az alhálózaton van, amelyik az IP címéből kideríthető. Mivel egy mobil eszköz akárhol is lehet, a vele való kommunikáció bonyolultabb szabályok szerint zajlik. Egy TCP kapcsolat fenntartásához az eszköznek állandó IP címmel kell rendelkeznie. Ez azt is jelenti, hogy egy mozgó eszköz nem dönthet egyszerűen úgy, hogy ő mindig

a legközelebbi adóállomás által kapott címet fogja használni. Mivel ez a kérdéskör az internetes címzést is érinti, nem orvosolható pusztán a hálózatelérési réteg szintjén; ennek protokollját ki kell valahogy bővíteni. Emiatt született meg a **Mobil IP** kiterjesztés, amelyről a 3220-as RFC-ben van szó.

A Mobil IP megoldja azokat a címzési kérdéseket, amelyeket az állandó IP címhez társuló másodlagos cím problémája vet fel. A 9.17 ábrán látható a Mobil IP hálózati környezet sémája. Az eszköz megőriz egy állandó címet az „otthoni hálózathoz". Egy speciális útválasztó, az otthoni hálózatban található „Otthoni Ügynök" (*Home Agent*) nyilvántart egy táblázatot, amely az eszköz pillanatnyi helyzetét az állandó címéhez társítja. Amikor az eszköz egy új hálózatba lép át, akkor regisztrálja magát egy „Idegen Ügynöknél" (*Foreign Agent*), hogy folytatni tudja a hálózati működést. Az Idegen Ügynök felveszi a mobil eszközt a látogatói listára, és az eszköz pillanatnyi helyzetére vonatkozó információt elküldi az Otthoni Ügynöknek. Az Otthoni Ügynök frissíti az erre a célra fenntartott kapcsolódási (*binding*) táblázatát: feljegyzi az eszköz pillanatnyi helyzetét. Amikor egy adatcsomag érkezik az eszköz számára az otthoni hálózaton, akkor ezt beillesztik egy másik csomagba, melyet elküldenek az idegen hálózatba, hogy kézbesítsék az ott tartózkodó eszköz számára.



9.17 ábra

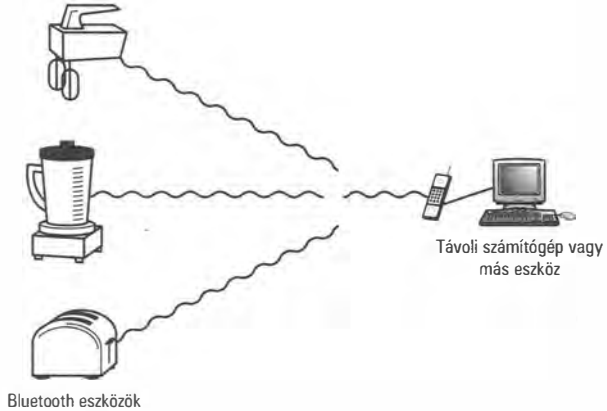
A WAP átjáró fordítja át a WAP protokollon érkező információt interneten is értelmezhető formátumúvá.

Bluetooth

A **Bluetooth** protokoll architektúra is a vezeték nélküli eszközök számára alakult ki; népszerűsége egyre nő a hálózati iparban. A Bluetooth-t az IBM és még néhány hasonló cég fejlesztette ki. A 802.11-hez hasonlóan a Bluetooth szabvány is az OSI adatkapcsolati és fizikai rétegeket definiálja (amely a TCP/IP hálózatelérési rétegével azonosítható).

A Bluetooth szabványt a legtöbbször olyan perifériákhoz használják, mint a fejhallgató vagy a vezeték nélküli billentyűzet/egér; ám a Bluetooth sok területen használható a 802.11-es szabvány helyett is. A Bluetooth rajongói mindig hangoztatják, hogy a 802.11 néhány biztonsági kérdését a Bluetooth már megoldotta. Az IBM hivatalos véleménye szerint azonban a Bluetooth és a 802.11 „egymást kiegészítő technológiák". Míg a 802.11 az Ethernet hálózat helyettesítésére született, a Bluetooth fő célja inkább az, hogy megbízható és nagy teljesítményű környezetet biztosítson a kis távolságban (tíz-egynéhány méteren belül) levő vezeték nélküli eszközök számára. A Bluetooth technológiát úgy tervezték, hogy megkönnyítse a kommunikációt egy kis területen (*Personal Area Network*; PAN) található néhány vezeték nélküli eszköz között.

Más vezeték nélküli megoldásokhoz hasonlóan a Bluetooth is egy hozzáférési pont révén kapcsolja a vezeték nélküli hálózatot a hagyományos hálózathoz. (Ennek a hálózati hozzáférési pontnak Network Access Point (NAP) a neve a Bluetooth szaknyelvben.) A Bluetooth Encapsulation Protocol (Bluetooth Egységbezáró Protokoll) betömöríti a TCP/IP csomagokat, hogy a Bluetooth révén kézbesüljenek a vezeték nélküli hálózatban.



9.18 ábra
Bluetooth-képes internet-híd

Ha egy Bluetooth eszköz elérhető az interneten keresztül, akkor elérhetőnek kell lennie a TCP/IP hálózat révén is. A gyártók már megálmodták az olyan internethez illeszkedő Bluetooth szabványt, amely révén az eszközök elérhetőek egy Bluetooth-képes internet-átjárón keresztül (9.18 ábra). Egy Bluetooth NAP hozzáférési pont viselkedhet hálózati hídként (*bridge*), amely TCP/IP adatokat kap, és az érkező adatcsomagok hálózatelérési rétegét Bluetooth hálózatelérési protokollréteggel helyettesíti, hogy az adatcsomagok eljuthassanak a rájuk várakozó eszközökhöz.

A nyelvészek el vannak ragadtatva attól, hogy a technológia létrehozói nem rövidítéssel jelölték művüket. De miért választották a Bluetooth (*„Kék fog"*) nevet? Az IBM részéről még érthető a „kékség", de honnan a fog? Talán megrágja az adatokat? Vagy mert bájtokat (szó szerint: falatokat) kap? Nem kell efféle metaforikus gyökeret keresnünk a kifejezés mögött. A Bluetooth név egy viking király nevéből ered: Harald Bluetooth Dániában és Norvégiában uralkodott a XI. században. Az ő idejében vette fel a kereszténységet az ország.

Bluetooth-t, a királyt sokan szerették, de törvényei meglehetősen önkényesek voltak. Az ő idejében élt egy „rossz fiú", aki a mintája lehetett Tell Vilmosnak. Egyszer azt parancsolta neki a király, hogy lője le az almát fia fejéről. Az íjász végrehajtotta a feladatot, de félretett három nyílvesszőt: ha a fiát bármi érné, azzal megölheti Bluetooth-t. Bízzunk benne, hogy vezeték nélküli világban az új Bluetooth által uralt eszközök nem keltenek efféle indulatokat, és nem vezetnek önbíráskodáshoz.

Hálózati kapcsolóelemek

Az előző órán kimerítően tárgyaltuk a TCP/IP hálózatokon működő útválasztók fontos témáját. Bár az útválasztók alapvető fontosságúak, és a szerepük igen jelentős, valójában azonban csak az egyik hálózati kapcsolóelemet jelentik a TCP/IP hálózatban.

Számos hálózati kapcsolóelem létezik – mindegyiknek megvan a maga szerepe a TCP/IP hálózati forgalom kezelésében. A következőkben a hidakról (*bridges*), elosztókról (*hubs*) és kapcsolókról (*switches*) lesz szó.

Hidak (bridges)

A **híd** olyan kapcsolóelem, amely a fizikai cím alapján szűri és továbbítja az adatcsomagokat. A hidak az OSI adatkapcsolati rétegében végzik tevékenységüket. Ahogy erről a 3. órában szó volt: ez a TCP/IP hálózatelérési rétegébe tartozik. Az utóbbi években ritkábban használnak hidakat a hálózatban, mivel rugalmasabb eszközök (például kapcsolók) léptek a helyükbe. A hidak egyszerű felépítése miatt azonban érdemes ezzel kezdenünk a hálózati elemek tárgyalását.

Bár a híd nem útválasztó, útválasztási táblázatot használ a kézbesítési információk kiderítéséhez. A fizikai címre alapozott útválasztó tábla meglehetősen eltér azoktól (vagyis jóval kevésbé kifinomult), mint amikről óránk későbbi részén szó lesz.

A híd figyeli az összes (vele összeköttetésben levő) alhálózatot, és felépít egy táblázatot arra vonatkozóan, hogy melyik fizikai cím melyik alhálózatban van. Amikor egy alhálózaton adattovábbítás zajlik, a híd megnézi a célállomás címét, és összeveti az útválasztó táblával. Ha a célállomás ugyanazon az alhálózaton van, mint ahonnan az adatcsomag érkezett, akkor a híd nem foglalkozik tovább az adatokkal. Ha azonban másik alhálózatba kell eljutnia a csomagnak, akkor továbbítja azt a megfelelő helyre. Ha a célállomás címe esetleg nincs benne az útválasztó táblában, akkor a híd az összes alhálózatnak továbbítja az adatcsomagot, kivéve azt, amelyikből érkezett.

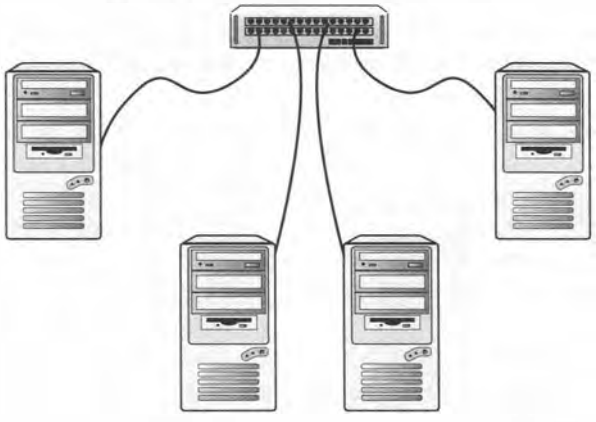
Fontos tudatosítani, hogy a hidak által használt hardveralapú fizikai címek nem azonosak a logikai IP címekkel. A kettő közti különbségről további információk találhatóak az 1-től 4-ig terjedő órák anyagában.

Egy időben igen elterjedten használtak hidakat a helyi hálózatokon, mivel olcsó megoldást biztosított a forgalom szűrésére, és így lehetővé vált, hogy egyre több számítógép bekapcsolódjon a hálózat működésébe. Ahogy óránkon már szó esett róla, olyan hálózat-elérési eszközökben is szerepet kap a híd funkciója, mint amilyen a kábelmodem és néhány DSL eszköz. Mivel a hidak csak a hálózatelérési rétegbe tartozó fizikai címe használják, és nem foglalkoznak az IP adatcsomag fejlécében elérhető logikai cím-információkkal, a hidak nem használhatóak eltérő hálózatok összekapcsolására. Alkalmatlanok az olyan nagyméretű hálózatokban történő IP útválasztásra és csomagkézbesítésre, mint amilyen az internet.

Elosztók (hubs)

Az Ethernet korai éveiben elterjedt volt az a gyakorlat, hogy az összes gépet egyetlen folyamatos koaxiális kábellel kapcsolták össze. Az utóbbi években inkább a 10BASE-T stílusú, elosztó alapú Ethernet hálózat terjedt el. Ma már szinte minden ilyen hálózatban van egy központi elosztó vagy kapcsoló, amelybe az összes számítógépet bekötik (9.19 ábra).

Emlékezzünk vissza a 3. órára, amelyben a klasszikus Ethernet fogalmát tárgyaltuk: az összes számítógép ugyanazon az átviteli csatornán osztozik. Minden üzenet eljut az összes hálózati csatolóhoz. Ha az Ethernet elosztó valamelyik „lábára" (kapujára) egy üzenet érkezik, akkor továbbküldi az összes többire (ez a helyzet látható a 9.19 ábrán is). Más szóval, a hálózat úgy viselkedik, mintha minden számítógép egyetlen folyamatos vezetékkel lenne összekötve. Az elosztó nem szűr és nem irányít adatokat, csak újraküldi, amit kap.



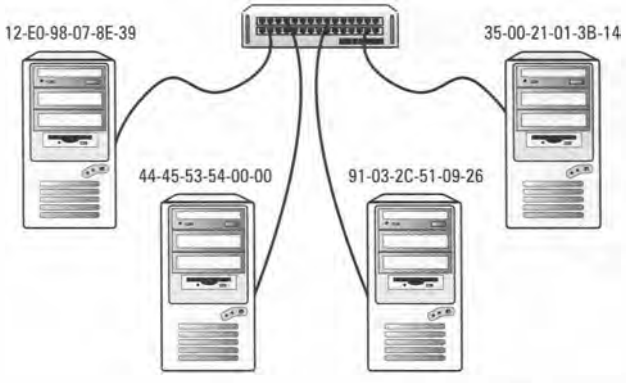
9.19 ábra
Elosztó alapú Ethernet hálózat

Az elosztó alapú Ethernet hálózatok azért terjedhettek el ilyen mértékben, mert nagymértékben leegyszerűsítik a hálózat huzalozását. Minden számítógépet egyetlen vezetékkel kapcsolódik az elosztóhoz. Így aztán bármelyik számítógép bármikor könnyedén le- vagy felcsatlakoztatható. Egy irodában, ahol sok gép van kis területen, egyetlen elosztó ki tud szolgálni számos gépet, sőt, (egyéb hálózatrészeket ellátó) más elosztókhoz is hozzákapcsolható. A gyártók hamarosan rájöttek, hogy ez a hálózati topológia (vagyis, hogy minden vezeték egyetlen kapcsolóelemből indul ki) számos újítási lehetőséget rejt magában. Kifinomultabb, úgynevezett **intelligens elosztók** jöttek létre, amelyek további lehetőségeket is biztosítottak, például a vonalhibák észlelését vagy egy-egy kapu kizárását. Az elosztók helyett ma már többnyire kapcsolókat használnak – ezekről pedig a következő szakaszban esik szó.

Kapcsolók (switches)

Az elosztó alapú Ethernet hálózat is örökölte mindazokat a hátrányokat, amelyek az Ethernetre általában jellemzőek: a forgalom növekedtével visszaesik a teljesítmény. A gépek csak akkor küldhetnek üzenetet, ha nem foglalt a vonal. Ráadásul minden hálózati csatolónak fogadnia kell (és fel kell dolgoznia) az összes adatkeretet, amely az Ethernetre el lett küldve. Az elosztók kifinomultabb változata a **kapcsoló** (*switch*), amely az Ethernet efféle problémáit hivatott orvosolni. A kapcsolók legalapvetőbb változatai hasonlítanak a 9.19 ábrán bemutatott elosztókra. Minden gép egyetlen kábellel csatlakozik a kapcsolóra. A kapcsoló azonban nem küldi ki ész nélkül az összes lábára az egyik irányból érkező adatcsomagokat. A legtöbb kapcsoló úgy dolgozik, hogy minden kapujához társít egy fizikai címet, mégpedig azét a hálózati csatolóét, amely az adott kapuhoz kapcsolódik (9.20 ábra). Amikor egy bekötött számítógépről adatkeret érkezik valamelyik kapura, a kapcsoló megvizsgálja a célállomás címét, és arra a kapura

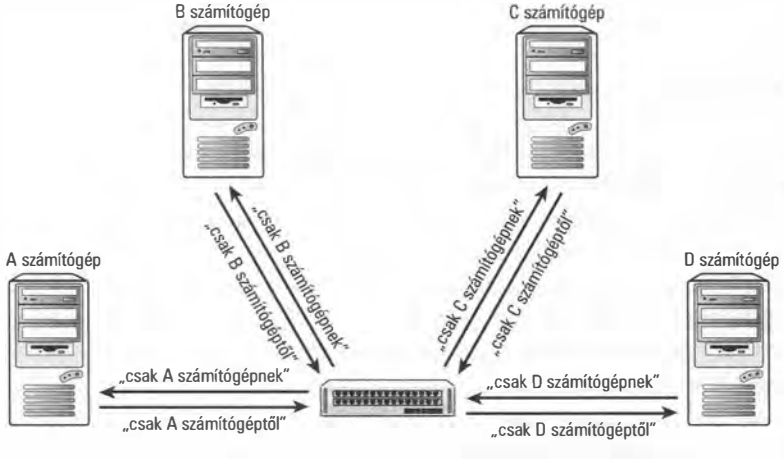
küldi tovább, amelyik az adott címhez tartozik. Röviden: a kapcsoló csak a címzett hálózati csatolónak küldi el az adatcsomagot. Nem kell minden csatolónak megvizsgálnia a hálózaton közlekedő összes keretet. A kapcsolók csökkentik a felesleges üzenetküldéseket, ezzel nagyban javítják a hálózat teljesítményét.



9.20 ábra

A kapcsoló minden kapuhoz társít egy fizikai címet

Figyeljük meg, hogy a fent vázolt kapcsoló fizikai címeket használ (3. óra), nem pedig IP címeket. A kapcsoló végtére is nem útválasztó. Sokkal közelebb áll a hídhoz – pontosabban olyan, mintha több hidat építettek volna egybe. A kapcsoló elkülöníti a hálózati kapcsolatait, tehát csak azok az adatok kerülnek rá a kábelekre, amelyek az adott számítógépnek vannak szánva (vagy onnan érkeznek), ahogy ez a 9.21 ábrán is látható.



9.21 ábra

A kapcsoló elszigeteli egymástól a különböző számítógépeket, így csökkenti a hálózati forgalmat

Manapság többféle kapcsoló is használatban van.

A két leggyakoribb kapcsolási módszer a következő:

- átvágó (*cut-through*) – Mihelyt megkapja a célállomás címét, a kapcsoló azonnal megkezdi a keret továbbítását.
- tárol-és-továbbít elvű (*store-and-forward*) – A kapcsoló megvárja, míg az egész keret megérkezik, és csak utána kezdi el a továbbítást. Ez némileg lassítja az újraküldési folyamatot, összességében mégis növelheti az eredő teljesítményt, mert így már maga a kapcsoló kiszűrheti a töredékes adatokat vagy az érvénytelen kereteket.

A kapcsolók rendkívüli népszerűségre tettek szert az utóbbi években. A vállalati hálózatok teljesítményének optimalizálására gyakran használnak egymásra rétegzett, összekötött kapcsolókat.

Vannak gyártók, akik az itt vázolt kapcsolási fogalmat csak egy tágabb értelemben vett kapcsolási elv speciális eseteként fogják fel. A kifinomultabb kapcsolók magasabb protokollrétegekben is tevékenykedhetnek, és így jobban paraméterezhető a csomagtovábbításra vonatkozó döntési mechanizmus. A csomagkapcsolás efféle tágabb értelmű fogalmában az eszközöket aszerint lehet rangsorolni, hogy melyik az a legmagasabb OSI protokollréteg, amelyben az adott eszköz tevékenykedik. Ebben a felfogásban a fejezetben vázolt kapcsoló (amely az OSI adatkapcsolati rétegében működik) **Layer 2** (2. rétegbeli) kapcsolónak minősül. Azokat a kapcsolókat, amelyek (az OSI hálózati rétegében található) IP cím információkra alapozva végzik munkájukat, Layer 3 kapcsolóknak hívják. Ahogy azt az olvasó nyilván sejti, a **Layer 3** kapcsolók lényegében sajátos útválasztók. Ha egy kapcsolóhoz nem neveznek meg ilyen réteg-paramétert, akkor alapértelmezetten a 2. rétegbeli kapcsolóelemeket szokták kapcsolónak hívni, amelyek fizikai (MAC) cím alapján dolgoznak – ahogy az fejezetünkben is előkerült.

Összefoglalás

Ezen az órán olyan technológiákról volt szó, amelyek révén megoldható az internethez (vagy más, nagyméretű hálózathoz) történő kapcsolódás. Tanultunk modemekről, ponttól-pontig (*point-to-point*) terjedő kapcsolatokról és a betárcsázós elérésről. Szó volt néhány népszerű szélessávú technológiáról (kábelmodemes és DSL hálózatokról), valamint vezeték nélküli technológiákról is. Megismertünk néhány fontos vezeték nélküli hálózati protokollt, és beszéltünk a TCIP/IP hálózatokon működő legfontosabb hálózati kapcsolóelemekről.

Kérdések és válaszok

K *A SLIP és PPP miért nem igényel olyan teljes fizikai címzési rendszert, mint amilyet az Ethernet hálózati kapcsolat?*

V Egy ponttól-pontig kapcsolat kialakításához nem szükséges (az Ethernethez hasonló) fizikai címzési rendszer, mert csak két számítógép vesz részt a kommunikációban; csak ők kapcsolódnak a vonalra. A SLIP és a PPP azonban teljes támogatást nyújt a logikai címzéshez (IP vagy akár más hálózatréteg-protokoll alapján).

K *A kábelmodemes kapcsolatom minden nap lelassul egy adott időszakban. Mi lehet a gond? Mit tehetek ez ellen?*

V A kábelmodem más eszközökkel megosztott átviteli közeget használ, így a teljesítmény lecsökkenhet a nagyfokú használat időszakában. Hacsak nincs lehetőségünk másik alhálózathoz kapcsolódni (ami elég ritka), akkor együtt kell élni a szélessávú kábelből eredő efféle problémával. Megpróbálhatunk átváltani DSL-re, amelyen a sávszélesség kevésbé függ egyéb paraméterektől. Kiderülhet azonban, hogy összességében a DSL nem gyorsabb a kábelnél – sok függ a szolgáltatás-minőség részleteitől, a helyi forgalmi szintektől és a körzetünkben elérhető szolgáltatóktól.

K *Miért van szükség arra, hogy egy mobil eszköz regisztrálja magát egy hozzáférési pontnál?*

V A hagyományos hálózat irányából érkező adatkeretek abból a hozzáférési pontból indulnak el a mobil eszköz felé, amelyiknél az regisztrálva van. A regisztrációkor arról tájékoztatjuk a hálózatot, hogy az adott mobil eszköznek szánt kereteket ennek a hozzáférési pontnak kell küldeni.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **802.11** – A vezeték nélküli kommunikációhoz kifejlesztett protokollkészlet. A 802.11 protokollok a TCP/IP verem hálózatelérési rétegét érintik, amely az OSI adatkapcsolati és fizikai rétegének felel meg.
- **Hozzáférési pont (*Access Point*)** – Olyan eszköz, amely összeköttetésül szolgál a hagyományos Ethernet hálózat és a vezeték nélküli hálózat között. A hozzáférési pontok általában hídként működnek: a kétfajta hálózatrész között közvetítik az adatkereteket.
- **Társítás** – Az a folyamat, amelynek során egy vezeték nélküli eszköz regisztrálja magát egy közeli hozzáférési pontnál.
- **Bluetooth** – Egymás közelében lévő vezeték nélküli készülékek protokoll-architektúrája.

- **Híd (*Bridge*)** – Hálózati kapcsolóelem, amely fizikai cím alapján továbbítja az adatcsomagokat.
- **Kábelmodem végberendezés (*Cable Modem Termination System; CMTS*)** – Olyan eszköz, amely lehetővé teszi, hogy kábelmodem kapcsolattal kapcsolódjunk a szolgáltató hálózatához.
- **Átvágó (*cut-through*) kapcsolás** – Olyan kapcsolási módszer, amelynél a kapcsoló azonnal megkezdi a keret továbbítását, mihelyt megkapja a célállomás címét.
- **Digital Over Cable Service Interface Specification (DOCSIS)** – A CableLabs non-profit szervezet által kifejlesztett protokoll, amelyet koaxiális kábelhálózatokon használnak internet, telefon és IPTV szolgáltatások biztosítására.
- **Digital Service Line Access Multiplexer (DSLAM)** – Olyan eszköz, amely lehetővé teszi, hogy DSL kapcsolattal kapcsolódjunk a szolgáltató hálózatához.
- **Digital Subscriber Line (DSL)** – Telefonvonalon megvalósított szélessávú kapcsolat.
- **Elosztó (*hub*)** – Hálózati kapcsolóelem, amelybe hálózati kábeleken keresztül kapcsolódnak egy alhálózat gépei. Általában nem végez csomagszűrést, csak (az összes kapura) továbbadja a kapott kereteket.
- **Független alap-szolgáltatáskészlet (*Independent Basic Service Set, IBSS*)** – Egymással közvetlenül kommunikáló eszközökből álló vezeték nélküli hálózat.
- **Infrastruktúrafüggő alap-szolgáltatáskészlet (*Infrastructure Basic Service Set, IBSS*)** – Egymással egy vagy több hozzáférési ponton keresztül kommunikáló eszközökből álló vezeték nélküli hálózat. A hozzáférési pontok egy hagyományos hálózathoz vannak kapcsolva.
- **Intelligens elosztó (*hub*)** – Olyan elosztó, amely az alapértelmezett elosztói tevékenységen kívül mást is végez, például azoknak a kapuknak a lezárását, amelynél vonalhiba észlelhető.
- **Link Control Protocol (LCP)** – A PPP által használt protokoll betárcsázós kapcsolat létrehozására, kezelésére és lezárására.
- **Maximum Receive Unit (MRU)** – Egy PPP keretbe zárt adatok maximális hossza.
- **Mobile IP** – IP címzési módszer, amelyet mozgó mobil eszközök támogatására fejlesztettek ki.
- **Modem** – Hálózati eszköz, amely digitális és analóg jelek között végez átalakítást (modulál/demodulál).
- **Network Control Protocol (NCP)** – Hálózatvezérlő protokoll, amelyet PPP felülethez terveztek, speciális protokollkészlettel.
- **Nyílt hitelesítés (*Open authentication*)** – Nyílt hitelesítési technológia, amelynél a hálózati elérést kérő eszközt egy előre beállított karakterlánc (Service Set Identifier; SSID) azonosítja.
- **Ponttól-pontig (point-to-point) kapcsolat** – (Pontosan) két kommunikáló eszköz között, megadott vonalon létrehozott adatkapcsolat.
- **Point-to-Point Protocol (PPP)** – Betárcsázós protokoll. A PPP támogatja a TCP/IP-t és más hálózati protokollkészleteket is. A PPP újabb és hatékonyabb, mint a SLIP.
- **Újratársítás** – Az a folyamat, melynek során egy vezeték nélküli eszköz megváltoztatja az eredetileg kijelölt hozzáférési pontját, és egy másikra regisztrálja magát.

- **Soros vonali internet protokoll (Serial Line Internet Protocol; SLIP)** – Kezdetleges TCP/IP alapú betárcsázós protokoll.
- **Megosztott kulcsos hitelesítés** – Olyan hitelesítési megoldás, amelynél minden eszköznek (a nyilvános kulcson kívül egy) titkos kulcs használatával kell tudnia bizonyítania azonosságát.
- **Tárol-és-továbbít elvű (*store-and-forward*) kapcsolás** – Olyan kapcsolási módszer, amelynek során a kapcsoló megvárja, míg az egész keret megérkezik, és csak utána kezdi el a keret továbbítását.
- **Kapcsoló (*Switch*)** – Hálózati kapcsolóelem. Ismeri az egyes kapuihoz tartozó címeket, így minden beérkező keretet a megfelelő kapuhoz tud továbbítani. A kapcsolók a protokollverem fejlécében található számos paramétertől függően hozhatnak meg csomagtovábbítási döntéseket.
- **Wide Area Network (WAN)** – Nagy kiterjedésű hálózat – olyan technológiák csoportja, amelyet nagy távolságok áthidalására, gyors és szélessávú kapcsolatok biztosítására terveztek.
- **Wired Equivalent Privacy (WEP)** – A 802.11 vezeték nélküli hálózatok biztonságos használatához tervezett szabvány.
- **Wireless Application Protocol (WAP)** – Vezeték nélküli alkalmazási protokoll – a protokollverem magasabb rétegeit érintő protokoll, melyet vezeték nélküli eszközökhöz terveztek.
- **Wireless Markup Language (WML)** – Vezeték nélküli jelölőnyelv – az XML leegyszerűsítése, melyet a WAP protokollokkal együtt használnak.
- **WAP Datagram Transport Protocol (WDP)** – WAP adatcsomag-átviteli protokoll. A WAP-os kapcsolat nélküli átviteli réteg protokollja, amely szerepét illetően hasonlít az UDP-hez (6. óra).
- **WAP Session Protocol (WSP)** – WAP munkamenet protokoll – a HTTP WAP-os megfelelője. A WSP biztosítja az alkalmazások közti adatcsere rendszerét.
- **WAP Transaction Protocol (WTP)** – WAP tranzakciós protokoll – a szolgáltatások közti kapcsolatfelvételt („kézfogást") és nyugtázást biztosító protokoll. Ezekkel lehet WAP tranzakciókat kezdeményezni vagy jóváhagyni.
- **WAP biztonsági tranzakciós réteg (WAP Transaction Layer Security ; WTLS)** – SSL alapú biztonsági protokoll (20. óra).

10. ÓRA

Tűzfalak

Ebben az órában a következőkről lesz szó:

- Mi az a tűzfal?
- A demilitarizált zóna (DMZ)
- Tűzfalszabályok
- A hagyományos és a fordított (*reverse*) proxy

A betörők tisztában vannak azzal, hogy a szerverek mindig kapcsolatot keresnek. A hálózatunkon futtatott összes szolgáltatás alkalmat szolgáltat a rossz fiúknak a betörésre. Nem tudunk azonban mindent leállítani. Mi más lenne egy hálózat célja, mint elősegíteni és támogatni a kommunikációt? Az évek során összegyűlt tapasztalatok alapján a szakértők arra az álláspontra jutottak, hogy a megoldás a következő: megfelelő védettségben érdemes üzemeltetni a hálózatunkat, a külső kapcsolatokat pedig csak jól körülhatárolt keretek között (megadott típusú kommunikációs csatornákon) szabad engedélyezni. Azt a speciális védőbástyát, amely lehetővé teszi a betörésektől mentes, védett hálózati teret, tűzfalnak hívjuk. Ezen az órán a tűzfalakról lesz szó a TCP/IP fényében.

Az óra anyagának elsajátítása után az olvasó képes lesz

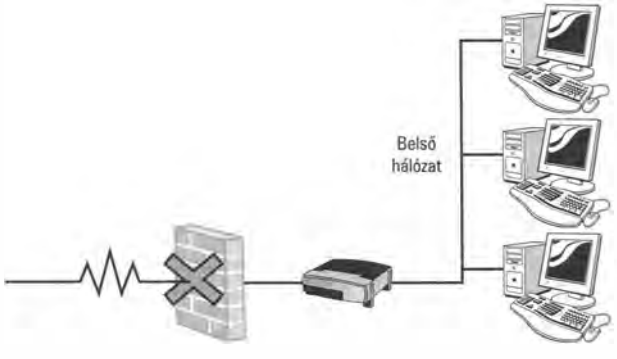
- vázolni, hogy mi az a tűzfal, és hogy mi a szerepe egy hálózatban
- Felsorolni különféle tűzfal-beállítási lehetőségeket
- Elmagyarázni a DMZ célját
- vázolni a proxy és a fordított proxy szolgáltatások előnyeit

Mi az a tűzfal?

A „tűzfal" kifejezés jelentése sokat változott az elmúlt években. Manapság már egy konkrét hardvereszközre szokás gondolni a tűzfal említésekor – ám ez hosszú fejlődés eredményeként alakult így. (28 év nagy idő a számítástechnikában.)

A tűzfal olyan eszköz, amelyet a hálózat határára szoktak tenni abból a célból, hogy (az adatcsomagok megfelelő továbbításával) elérhetővé tegye a „kívülállók" számára a hálózatot. Első hallásra ez a megfogalmazás az útválasztó fogalmára emlékeztet. Bár a tűzfalnak nem kell útválasztónak is lennie, az útválasztókba gyakran beépítenek tűzfal funkciókat. Az alapvető különbség az, hogy a (klasszikus értelemben vett) útválasztó mindig továbbítja adatcsomagot, amikor *tudja* – a tűzfal pedig csak akkor, amikor *akarja*. A továbbításra vonatkozó döntés nem csak a rendeltetési címtől függ (mint az útválasztónál), hanem különféle szabályoktól. Ezeket a hálózat tulajdonosa hozza meg attól függően, hogy milyen fajta forgalmat szeretne beengedni a hálózatába.

A tűzfal szerepe azonnal világossá válik, ha ránézünk a 10.1 ábrára. Ez egy igen egyszerű vázlat, de jól mutatja, hogy a tűzfal olyan helyzetben van, hogy útját tudja állni a külvilágból érkező forgalomnak, de nem szól bele a hálózat gépeinek kommunikációjába.



10.1 ábra
A tűzfal akadályozni tudja a belső hálózat felé irányuló forgalmat

Az első tűzfalak **csomagszűrők** voltak. Megvizsgálták az adatcsomagokat, hogy található-e valami jele a rosszindulatú adatforgalomnak. Ahogy a 6. órán („A szállítási réteg") tanultuk, sok csomagszűrő tűzfal a jól ismert TCP és UDP kapuszámokat figyeli, melyek a szállítási réteg fejlécében szerepelnek. Mivel a legtöbb internetes szolgáltatáshoz

konkrétan adott kapuszám tartozik, nagy valószínűséggel meg lehet állapítani egy csomag rendeltetését a kapuszám alapján. A csomagszűrésnek ez a változata lehetővé teszi a rendszergazdák számára, hogy azt mondják: „külső ügyfeleink nem használhatják a belső hálózaton működő telnet szolgáltatást" – legalábbis abban az esetben, ha a belső telnet szolgáltatásunk a közismert telnet kaput használja.

Ez a fajta felügyelet kiválóan működött régebben, és ma is távol tartja a támadások nagy részét. Be kell azonban vallanunk, hogy a csomagszűrés nem jelent teljes körű megoldást. Többek között azért sem, mert ha egy támadó bejut a belső hálózatra, akkor átállíthatja a hálózati szolgáltatásokhoz tartozó kapuszámokat. Ha a tűzfal úgy van beállítva, hogy a 23-as TCP kapun figyelje a telnet kapcsolatfelvételi kéréseket, akkor a támadó megteheti azt, hogy felállít egy titkos telnet szolgáltatást, amely egy másik kapuszámon működik. Így már hatástalan lesz a jól ismert kapuszámok figyelése.

A tűzfal fejlődésének következő állomása az úgynevezett **állapot alapú (*stateful*) tűzfal** volt. Az állapot alapú tűzfal nemcsak önmagában nézi az egyes csomagokat, de azt is figyeli, hogy a csomag hol kapcsolódik be a kommunikációs munkamenetek sorozatába. A „kommunikációs állapotokra" vonatkozó ilyetén érzékenység teszi alkalmassá az állapot alapú tűzfalat arra, hogy további trükközéseket is kiszűrjön, mint például az érvénytelen csomagok küldése, kapcsolat-eltérítési (*session hijacking*) kísérletek, és a különféle túlterheléses szolgáltatás-megtagadási (*DoS*) támadások.

Az új generációs, alkalmazási rétegbeli tűzfalaknak (az alkalmazási réteg adataihoz is hozzáférve) sokkal teljesebb körű rálátásuk lehet az adott csomagokhoz tartozó protokollokhoz és szolgáltatásokhoz.

A mai tűzfalak általában mind csomagszűrő, mind állapotfigyelő és alkalmazási réteget is használó szolgáltatást is nyújtanak. Vannak tűzfalak, amelyek mindezek mellett még DHCP szerverként és hálózati címfordító (NAT) eszközként is szolgálnak. A tűzfalak szoftverek is lehetnek, de célhardverek is készülnek – vannak egyszerűbbek és vannak kifinomultabbak. Azonban akár egyetlen számítógépet szeretnénk megóvni, akár egy több száz gépet tartalmazó hálózatot, érdemes legalább alapszintre eljutni a tűzfalakkal kapcsolatban, ha az internet közelébe szeretnénk kerülni.

Tűzfal-beállítási lehetőségek

A tűzfalakat régebben az informatikai szakértők szakterületének tekintették, manapság már a hálózati betörések és az automatikus kaputapogatók elterjedésével megváltozott a helyzet. Az automatikus kapuletapogatók véletlenszerűen keresnek az interneten nyitott kapukat, így az egyfelhasználós rendszerekhez is szükségessé vált a személyi tűzfalak kialakítása. Sok mai Windows, MacOs és Linux rendszeren van személyi tűzfal, amelyet arra terveztek, hogy a rendszer egyes kapuit és szolgáltatásait megvédje. Egy végfelhasználói rendszernek természetesen nem kell hálózati szolgáltatások sokasá-

gát futtatnia, így látszólag felesleges is tűzfalat használni („...miért is zárnánk ki olyan szolgáltatásokhoz tartozó kapukat, amelyek első megközelítésben nem is futnak?"). Az igazság az, hogy a modern számítógépes rendszerek már annyira összetettek lehetnek, hogy nem mindig lehetünk biztosak abban, hogy mi fut és mi nem. A számítógépeknek számos sebezhető pontja lehet, így nem mindig könnyű meggyőződni arról, hogy rendszerünk valóban teljesen biztonságos-e. Jó ötletnek tartom a személyi tűzfalakat – különösen olyan gépek esetén, amelyek nem állnak más tűzfal védelme mögött.

Az egyéni számítógépek után a következő bonyolultsági szintet a SOHO hálózatok (*small office, home office – kis iroda, otthoni iroda*) jelentik – ezekhez is szükség van tűzfalra és útválasztóra. Ezek az eszközök általában DHCP szolgáltatást és hálózati címfordítást (NAT) is nyújtanak. Általában a 10.1 ábrán is bemutatott klasszikus helyzetre tervezték őket, vagyis a belső gépek szabadon elérhetik a belső hálózat szolgáltatásait, de a külső gépek számára ezek a szolgáltatások nem elérhetőek.

Az előregyártott SOHO (és időnként a személyi) tűzfal-hardverekkel az a gond, hogy laikusok számára tervezték őket, így nem sok beállítási lehetőséget kínálnak föl. Sokszor még az sem derül ki, hogy milyen módszer alapján szűrik a protokollokat, a forgalmat. A biztonsági szakértők ezeket az eszközöket nemigen tekintik teljesen megbízhatónak, bár nyilván jobb használni őket, mint tűzfal nélkül dolgozni.

Egy másik lehetőség egy normál számítógép használata tűzfalként/útválasztóként. A Unix/Linux rendszerekhez igen kifinomult tűzfalprogramok állnak rendelkezésre. Néhány Windows változathoz is elérhetőek tűzfalak. Meg kell jegyeznünk azonban, hogy egy tűzfalként üzemeltetett önálló számítógép nem csak arra képes, mint a szakaszunkban korábban említett személyi tűzfal. Ilyenkor a tűzfalként működő számítógép nemcsak megszűri a neki címzett csomagokat, hanem ténylegesen úgy működik, mint a hálózat tűzfala. Ehhez rendelkeznie kell legalább két hálózati kártyával és megfelelően be kell rajta állítani a forgalom-továbbítást – végső soron úgy kell működnie, mint egy útválasztónak. Ha van raktáron egy felesleges számítógép, akkor ily módon sokkal kifinomultabb tűzfalmegoldások érhetőek el, mint egy dobozos SOHO tűzfallal. Természetesen mindig tudnunk kell, hogy mit csinálunk.

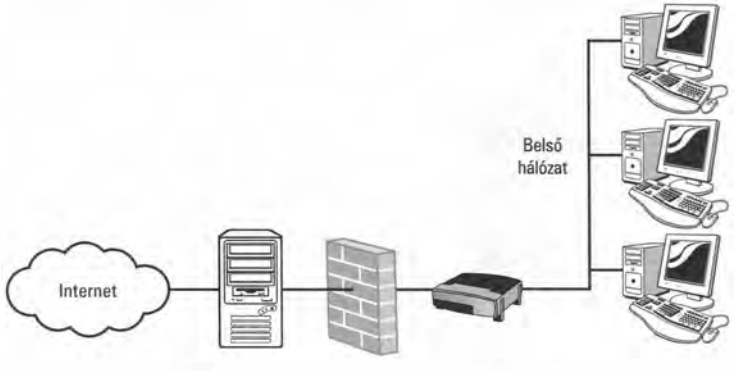
Ha profi tűzfalmegoldások kell adminisztrálnunk, akkor nyilván valamilyen kereskedelmi tűzfal-hardvereszköz van a kezünk ügyében. A profi szintű tűzfalak/útválasztók sokkal kifinomultabbak, mint a SOHO tűzfal-hardverek. Belülről igencsak hasonlítanak egy (tűzfalként üzemeltetett) számítógéphez, bár kívülről ez nem mindig látszik. A legtöbb ipari tűzfal egy saját operációs rendszerbe van beépítve. Ahogy tanulni fogjuk: a kereskedelmi tűzfalak és a tűzfal-célszámítógépek lehetővé teszik tetszőleges szűrőszabályok megadását, amelyek alapján a különféle forgalomtípusokat kizárhatjuk vagy engedélyezhetjük. Ezek az eszközök sokkal hatékonyabbak és sokoldalúbbak, mint amit egy SOHO (vagy személyi tűzfal) beállítására szolgáló néhány jelölőnégyzettel meg lehet oldani. Az is igaz persze, hogy használatukhoz és megfelelő beállításukhoz alaposabb ismeretek (és nagyobb figyelem) szükséges.

10

A demilitarizált zóna (DMZ)

A tűzfal védett teret biztosít a belső hálózat számára, így ezt nem könnyű kintről elérni. Ez a forgatókönyv jól működik akkor, ha a belső hálózaton csak olyan munkacsoportok léteznek, amelyben csak néhány webes ügyfélprogram és néhány kósza fájlszerver található. Egy szervezet azonban nem mindig akarja az összes erőforrását elzárni a külvilág elől. Elképzelhető, hogy fut egy nyilvános webkiszolgáló; azt szeretnénk, hogy ez a hálózaton kívülről is látsszon. Sok szervezet FTP kiszolgálót is fenntart, esetleg levelezőszervereket vagy több más hasonló hálózati alkalmazást is, amelyeket az interneten keresztül is jó volna elérni. Elméletileg megoldható, hogy kaput nyissunk a tűzfalon, amelyen egy adott rendszer egy adott szolgáltatásához lehet hozzáférni kintről. Így lehetővé válna, hogy egy tűzfalon belüli kiszolgáló kívülről származó forgalmat fogadjon, ám ez óriási többletterhelést okozna és különféle biztonsági kérdéseket is felvetne, amelyet a legtöbb adminisztrátor szeretne elkerülni.

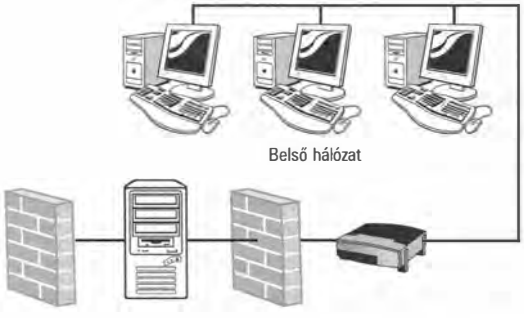
Kézenfekvő az elgondolás, hogy az internetről is elérendő szolgáltatásokat helyezzük ez a tűzfalon kívülre (10.2 ábra). Az ötletet még azzal lehet némileg javítani, hogy a (web)kiszolgálót vessük alá alapos vizsgálatnak, hogy valóban biztonságos-e, és ezután már kitehetjük az internet veszélyeinek. Ily módon a tűzfal már csak a belső hálózat felhasználóitól választja el, a külső felhasználóktól nem. Elvileg be lehet úgy állítani egy kiszolgálót, hogy képes legyen kivédeni az internetről érkező támadásokat. Csak a használandó kapukat kell nyitva tartani, és csak az életbevágóan fontos szolgáltatásokat szabad futtatni. Egy biztonságos rendszer jól meg van tervezve, így ha egy támadó elérést tud is szerezni a rendszerhez, a jogosultságai korlátozottak. Természetesen az efféle feltételezések nem zárhatják ki a rendszer feltörését. Az ötlet lényege azonban az, hogy ha a betörő meg is szerzi a webkiszolgáló felett a hatalmat, akkor sem tud bejutni a belső hálózatra: a tűzfal ezt (jó eséllyel) megakadályozza.



10.2 ábra

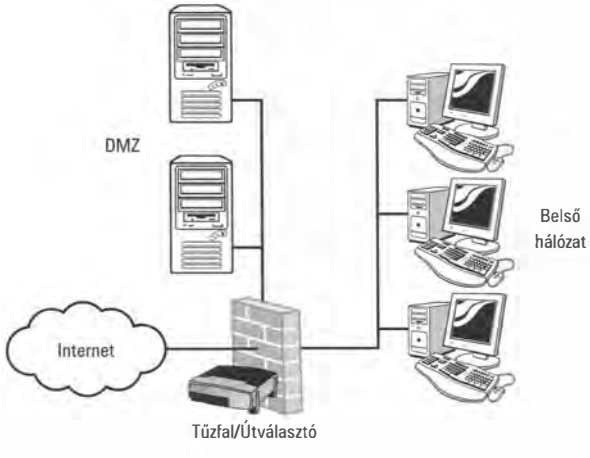
A webkiszolgálókat és más internetes szolgáltatásokat a tűzfalon kívül is el lehet helyezni.

A helyi erőforrások tűzfalon kívül történő elhelyezése nem ritka megoldás kisebb hálózatok esetén. Egy professzionális információtechnológiai vezetőkkel és biztonsági házirenddel rendelkező nagyobb hálózatnál azonban más megoldást szoktak választani. A 10.3 ábra két tűzfallal rendelkező hálózatszerkezetet mutat; az egyik az internet kiszolgáló gép előtt, a másik pedig emögött található. Az első tűzfal biztosít egyfajta biztonsági réteget, amely azonban eléggé átjárható ahhoz, hogy a kiszolgálókhoz érkező kapcsolatok kiépülhessenek. A hátrébb levő tűzfal biztosítja a tűzfalaktól megszokott nagyobb fokú szigorúságot; ez biztosítja a helyi hálózat erőforrásainak védelmét. A két tűzfal közti részt (némi utalással a vietnami háború katonai szóhasználatára) demilitarizált zónának, DMZ-nek hívják. A DMZ biztosítja azt a köztes biztonsági szintet, amely ugyan megbízhatóbb, mint a nyílt internet, de itt még nem érvényes a belső hálózat szigorúsága.



10.3 ábra
A DMZ a két tűzfal közötti hálózatrész.

Előfordulhat, hogy a 10.3 ábrán látható megoldáshoz hasonlót egyetlen tűzfallal valósítanak meg. Ilyenkor ugyanahhoz a tűzfalhoz kapcsolódik mindkét alhálózat. Ahogy az a 10.4 ábrán is látszik, a három (vagy több) hálózati csatlakozóval rendelkező tűzfal/útválasztó össze tudja kapcsolni a DMZ-t és a belső hálózatot, csak megfelelő szűrőszabályokat kell megadni az egyes felületekhez.

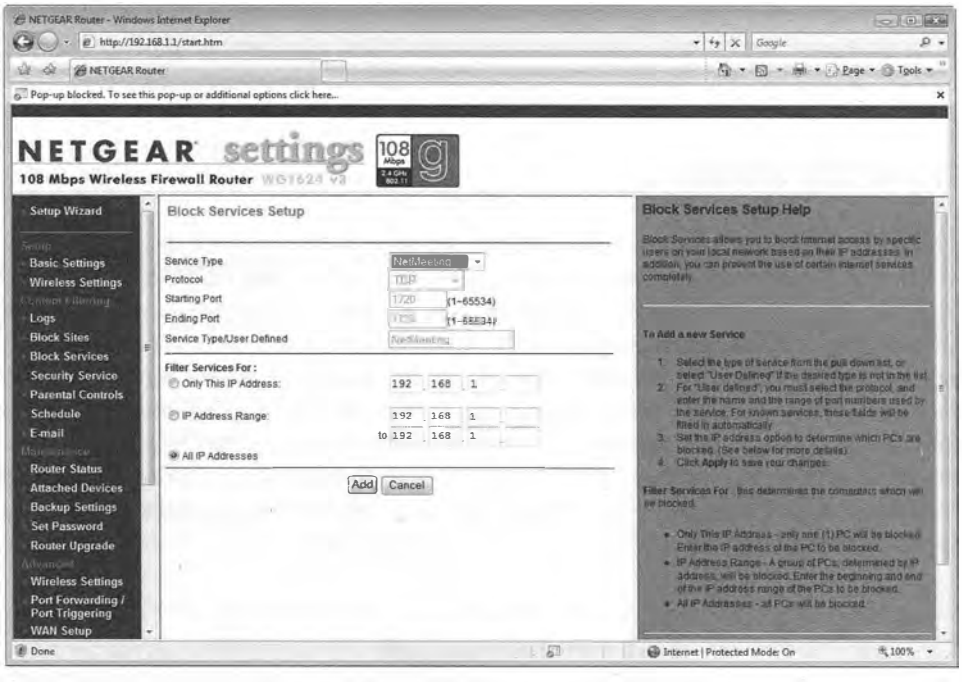


10.4 ábra
Egyetlen (legalább három „lábbal" rendelkező) tűzfal is meg tudja valósítani a DMZ-t, ha minden belső hálózatrészhez alkalmas szabályokat adunk meg.

10

Tűzfalszabályok

A személyi tűzfalak (és a hozzájuk hasonló egyszerű, grafikus felületen beállítható tűzfalszerű védelmi eszközök) beállításakor jelölőnégyzetek segítségével adhatjuk meg a kívánt szűrési szabályokat (10.5 ábra).



10.5 ábra

A legtöbb SOHO tűzfal név vagy kapuszám alapján teszi lehetővé a szolgáltatások szűrését.

A teljes körű védelmet nyújtó, ipari szintű tűzfaleszközökhöz azonban létrehozhatunk egy konfigurációs fájlt, amelyben megfelelő tűzfalbeállító kifejezésekkel, utasításokkal adhatjuk meg a kívánt viselkedést. Ezeket a parancsokat vagy szabályokat **tűzfalszabályoknak** (*firewall rules*) hívjuk. A különféle eszközök eltérő parancsokat és szintaxist használhatnak, de a tűzfalszabályok szinte minden esetben a következő fogalmakra hivatkoznak paraméterként:

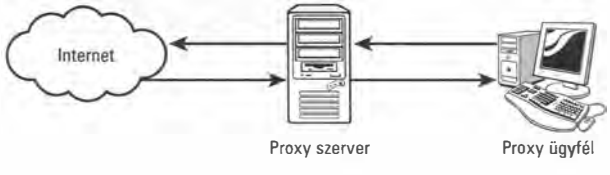
- Az adatcsomagok forráscíme vagy címtartománya
- A rendeltetési hely címtartománya
- Szolgáltatás
- Művelet (*action*)

Ezek a paraméterek igencsak kiszélesítik mozgásterünket a beállítási lehetőségek vonatkozásában. Letilthatóak adott címtartományokból (vagy akár bárhonnan) érkező konkrét szolgáltatás-kérések, mint például a telnet vagy az FTP. A művelet lehet elfogadás (*accept*), megtagadás (*deny*) vagy akár más opciók is használhatóak. A szabályok hivatkozhatnak egy adott kiterjesztésre vagy szkriptre is, aminek a tartalma lehet például egy riasztási parancs, amely a tűzfal-adminisztrátor számára emailt (vagy/és a telefonjára SMS-t) küld probléma esetén.

Ezeknek a paramétereknek a kombinációi sokkal több mindenre lehetőséget adnak, mint ami kapuszámmal megadott szolgáltatások ki-be kapcsolásával elérhető.

Proxy szolgáltatás

A tűzfal számos technológiai ötletet egyesít, amelyek abból a célból születtek, hogy védjék (és egyszerűen adminisztrálhatóvá tegyék) a belső hálózatot, és határok közé szorítsák az internetről érkező, megjósolhatatlan és esetleg veszélyes tevékenységek hatásait. Egy ezekkel rokon technológiát jelent a proxy szolgáltatás. A **proxy szerver** fogadja az ügyféltől az internetes erőforrásokra vonatkozó kéréseket, majd az ügyfél nevében küld szolgáltatási kérést a tényleges kiszolgálónak. Úgy működik, mint egy köztes eszköz az ügyfél és a kiszolgáló között (amelyhez az eredeti kérést küldték; 10.6 ábra). Bár a proxy szerver nem feltétlenül elég önmagában a hálózat védelméhez, gyakran használják tűzfallal együtt. Különösen akkor, ha mindezek a hálózati címfordítás (NAT) környezetében vannak, amiről a 12. órában lesz szó.



10.6 ábra

A proxy szerver a kliens nevében küld szolgáltatási kéréseket

Azáltal, hogy az ügyfél helyett a proxy szerver intézi az internetes szolgáltatás-kérések küldését, védi az ügyfelet attól, hogy közvetlen kapcsolatba kerüljön megbízhatatlan (és potenciálisan rosszindulatú) webes erőforrásokkal. Vannak olyan proxyk, amelyek egyfajta tartalomszűrést is végeznek: figyelik a feketelistára került kiszolgálókat és az esetleg veszélyes tartalmakat. A proxy szerverek arra is felhasználhatóak, hogy határok közé szorítsák a belső hálózat ügyfeleinek böngészési lehetőségeit. Egy iskolai hálózat proxy szervere például megakadályozhatja a gyerekeket abban, hogy nem nekik való oldalakat nézzenek.

A legtöbb esetben azonban a proxy szervert sokkal inkább teljesítményjavító hatásáért használják, nem pedig a biztonsági szempontok miatt. A proxy szerverek gyakran végeznek **tartalom-gyorstárazást**. A tartalmat gyorstárazó proxy szerver eltárolja azokat a weboldalakat, amelyek áthaladnak rajta. Az ilyen oldalakra vonatkozó későbbi kérések így helyileg is kiszolgálhatóak, sokkal gyorsabban, mint ha újra az internetről kellene lekérni őket. Első látásra felesleges felhajtásnak tűnik mindez pusztán azért, hogy segítsük azokat a felhasználókat, akik kétszer nézik meg ugyanazt az oldalt. De ha megfigyeljük egy átlagos felhasználó böngészési szokásait, akkor az derül ki, hogy meglehetősen gyakran kattintgatnak egy webhely környékén, és egy-egy oldalt többször is megnéznek. Az is előfordul, hogy elhagynak egy oldalt, majd rövid idő múltán újra visszatérnek ugyanoda. A proxy szervert általában úgy állítják be, hogy egy megadott időtartamra tárazza csak be a tartalmakat – ha ez letelik, akkor az adott oldalból friss verziót kéri a webszervertől.

Fordított (reverse) proxy

A(z előző szakaszban bemutatott) hagyományos proxy szerver a hálózatból kifelé irányuló internetes kérések kiszolgálójaként működik. A proxy szerver másik változata a **fordított (*reverse*) proxy**, amely külső helyekről kap kéréseket, és a belső hálózatra küldi őket. A fordított proxy ugyanazt a tartalom-gyorstárazást és tartalomszűrést tudja biztosítani, mint amit a hagyományos proxy szerver. Mivel a fordított proxykat elsősorban olyan számítógépekhez használják, amelyek az internet számára kínálnak különféle szolgáltatásokat, a biztonsági megfontolások ilyen esetben különösen fontosak.

A fordított proxy elrejti annak a számítógépnek a jellemzőit, amely ténylegesen kiszolgálja az ügyfelek kéréseit. Van teljesítményjavító hatása is, a nagy állományok vagy gyakran kért oldalak gyorstárazása révén. A fordított proxykat időnként terhelés-kiegyenlítésre használják. Megoldható például, hogy egy bizonyos weboldal lekéréseit egy adott fordított proxy kapja meg, de ő ezeket a kéréseket több szerver használatával szolgálja ki.

Összefoglalás

Ma már nem képzelhető el egy modern hálózat tűzfal nélkül – ez lehet egy célhardver, de lehet egy alkalmazás is. A tűzfal figyeli a bejövő forgalmat, és kizárja a gyanús adatcsomagokat. A tűzfalak a kimenő forgalmat is szűrhetik, ha érvényesíteni kell a testületi házirendet; ezen kívül meggátolhatják a kockázatos webhelyekhez történő hozzáférést. Ezen az órán többféle tűzfalról is szó volt. Megismerkedtünk a tűzfalszabályok fogalmával és vázoltuk a proxy (és fordított proxy) szolgáltatások előnyeit.

Kérdések és válaszok

K *Mi az állapot alapú tűzfal előnye?*

V A kapcsolatok állapotának vizsgálata révén az állapot alapú tűzfal felfedhet szolgáltatás-megtagadási kísérleteket, valamint kiszűrheti az érvénytelen csomagokat és észrevehet olyan trükközéseket, amelyek el akarják téríteni (vagy manipulálni akarják) a munkameneteket.

K *Mi a DMZ célja?*

V A DMZ célja egy köztes biztonsági zóna biztosítása, amely elérhetőbb a belső hálózatnál, de védettebb a nyílt internetnél.

K *Hogyan javítja a proxy szerver a webböngészők válaszidejét?*

V Sok proxy szerver gyorstárazza a korábban megnézett weboldalakat. Ezzel a tartalom-gyorstárazási módszerrel lehetővé válik, hogy bizonyos oldalakat ne a távoli (és így időigényes) internetes webkiszolgálóról, hanem a gyors helyi rendszerből vegyen a webböngésző.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **DMZ** – Köztes tér internetes szolgáltatásokat nyújtó szerverek számára. Ezeket a homlokzati tűzfal mögé, de egy szigorúbb (a belső hálózatot védő) másik tűzfal elé szokták üzembe állítani.
- **Tűzfal** – Olyan célhardver vagy alkalmazás, amely szűkíti a hálózati hozzáférést a belső hálózathoz.
- **Csomagszűrő** – Olyan tűzfal, amely kapuszám vagy más (az adatcsomag rendeltetésére utaló) protokoll-információ alapján végzi a szűrést.
- **Proxy szerver** – Olyan számítógép vagy alkalmazás, amely egy kliens nevében küld el kéréseket egy szervernek.
- **Fordított proxy** – Olyan számítógép vagy alkalmazás, amely a belső hálózat felé irányuló kéréseket fogad az internetről, és ezeket továbbítja a belső szervernek.
- **Állapot alapú tűzfal** – Olyan tűzfal, amely egy kapcsolat állapotát figyeli.

11. ÓRA

Névfeloldás

Ebben az órában a következőkről lesz szó:

- Gépnevek feloldása
- DNS
- NetBIOS

A második órán, a *„Hogyan működik a TCP/IP"* című részben már tanultunk a névfeloldásról. Ez egy igen hatékony megoldás arra, hogy (az emberek számára jobban kezelhető) alfanumerikus nevet kapcsoljunk a 32 bites IP címekhez. A névfeloldás folyamatában kapunk egy számítógépnevet, és megpróbáljuk megkeresni a hozzá tartozó IP címet. Ezen az órán gépnevekről, tartománynevekről és teljes tartománynevekről (*Fully Qualified Domain Name, FQDN*) lesz szó. Egy alternatív névfeloldási rendszerről, a NetBIOS-ról is beszélünk, amelyet gyakran használnak Microsoft hálózatokban.

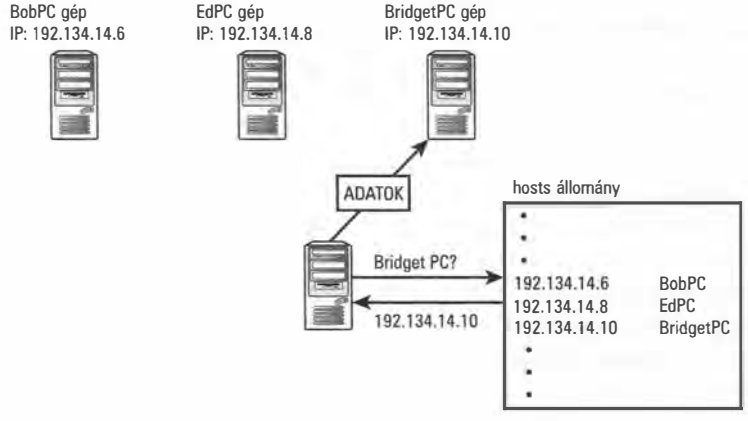
Az óra anyagának elsajátítása után az olvasó képes lesz

- elmagyarázni, hogy hogyan működik a névfeloldás
- különbséget tenni a gépnevek, tartománynevek és teljes tartománynevek között
- vázolni a gépnevek feloldását
- vázolni a DNS névfeloldást
- vázolni a NetBIOS névfeloldást

Mi az a névfeloldás?

Amikor az első TCP/IP hálózatok működni kezdtek, a felhasználók hamar érzékelték, hogy nem könnyű (és nem hatékony) fejben tartani a hálózat összes gépének IP címét. A kutatóközpontok szakemberei túl elfoglaltak voltak ahhoz, hogy emlékezzenek arra, hogy a 6. épület „A" számítógépének 100.12.8.14 vagy 100.12.8.18 az IP címe. A számítógépes szakemberek mindig lehetőséget keresnek a gépies teendők automatizálásra. Ha egy programozónak bármit is le kell írnia papírra, akkor biztosak lehetünk abban, hogy már töri is a fejét: hogyan lehetne olyan megoldást találni, hogy ezt az információt közvetlenül a számítógépnek lehessen átadni, és őneki minél kevesebb dolgot kelljen megjegyezni? Itt is ez merült fel: jobb volna, ha a gép párosítaná a címekhez a neveket.

A gépnevek rendszere egy egyszerű névfeloldási módszer, amely a TCP/IP hálózatok hőskorában alakult ki. Minden számítógép kapott egy alfanumerikus nevet, amelyet **gépnévnek** (*hostname*) hívnak. Ha az operációs rendszer IP címet vár valahol (például egy parancsban), és helyett efféle alfanumerikus névbe botlik, akkor megnézi a hosts állományt (*gépek*; lásd a 11.1 ábrát). A **`hosts` állomány** egy listát tartalmaz, amelyben gépnevek vannak IP címekhez párosítva. Ha az alfanumerikus név megtalálható itt, akkor a számítógép a hozzá tartozó IP címet fogja használni a gépnév helyett ott, ahol a fenti parancsban IP címet várt; így hajtja végre a parancsot.



11.1 ábra
A gépnevek feloldása

Az egyszerű gépnevek itt vázolt rendszere jól működött (és működik a mai napig is) kis hálózatokban. Ahogy azonban egyre nagyobb hálózatok jöttek létre, más megoldást kellett keresni. A géphez-cím párosításoknak egyetlen fájlban kell elférniük, így a keresés hatékonysága a fájl méretének növekedtével arányosan romlik. Az ARPAnet régi napjaiban egyetlen `hosts.txt` mesterállományban kellett nyilvántartani az összes

nevet és címet – ennek karbantartását a helyi adminisztrátorok végezték. A névtér alapvetően egydimenziós (és minden csomópont egyenértékű) volt. Ez a névfeloldási rendszer nem tudta kihasználni az IP címek névterének hierarchikus struktúráját.

Bár az ARPAnet rendszermérnökeinek sikerült felülemelkedni a problémákon, az egyszerű gépnevek efféle rendszere nem lenne működőképes a több milliónyi csomópontot tartalmazó internethálózatban. A mérnökök tisztában voltak azzal, hogy olyan hierarchikus névfeloldási rendszerre van szükség, amely...

- a névfeloldás feladatát többfelé osztja speciális a névfeloldó kiszolgálók között. A névfeloldó kiszolgálókon találhatóak meg azok a táblázatok, amelyek a gépnevek és címek egymáshoz rendelését tartalmazzák.
- a helyi névfeloldás feladatát és jogkörét a helyi adminisztrátorra bízza. Más szóval: ahelyett, hogy fenntartanánk egy központosított mesterverziót a név-cím párokból, meg kell oldani, hogy az „A" hálózat adminisztrátora kezelhesse az „A" hálózat névfeloldási teendőit, a „B" hálózat adminisztrátora pedig a „B" hálózatéit stb. Ily módon a hálózat változásaiért felelős szakemberek lesznek felelősek azért, hogy a hálózat változásai tükröződjenek a névfeloldási rendszerben is.

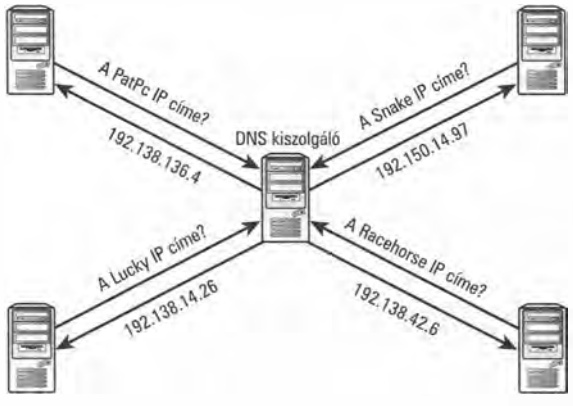
Ezek a szempontok vezettek el a DNS névfeloldási rendszerhez. A DNS névfeloldási rendszerét használja a teljes internet – ebből adódnak az afféle nevek, mint a *www.unixreview.com* és a *www.slashdot.org*. Ahogy azt hamarosan megismerjük: a DNS hierarchikus egységekre, úgynevezett **tartományokra** (*domains*) osztja a névteret. A tartománynév egybefoglalható a gépnévvel; így kapjuk a **teljes tartománynevet** (FQDN, *Fully Qualified Domain Name*). A whitehouse.gov tartományban működő maybe nevű gépnek például *maybe.whitehouse.gov* a teljes tartományneve.

Ebben az órában a gépnévfeloldásról és a DNS névfeloldási rendszerről lesz szó. Tanulunk a NetBIOS-ról is, amely a Microsoft hálózatok némelyikében használt névfeloldási megoldás.

A DNS névfeloldás

A DNS tervezői el akarták kerülni azt, hogy minden számítógépen karban kelljen tartani egy névfeloldási adatállományt. Ehelyett néhány (vagy egyetlen) kiszolgálóra bízták az adatok tárolását. A DNS kiszolgálók biztosítják a névfeloldási szolgáltatásokat az egész alhálózat számára (11.2 ábra). Ha az alhálózat egyik számítógépe gépnevet talál egy olyan parancsban, ahol IP címet várna, akkor elküld egy lekérdezést a kiszolgálónak, hogy megtudja a hiányzó IP címet. Ha a DNS kiszolgáló ismeri a keresett címet, akkor visszaküldi azt. A lekérdezést indító számítógép ezután kicseréli a szóban forgó gépnevet az IP címre, amivel már végre tudja hajtani a parancsot. Amikor megváltozik valami a hálózaton (például új számítógépet kapcsolnak be, vagy valamelyiknek megváltozik

a neve), akkor a hálózati adminisztrátornak csak egy helyen kell megváltoztatnia a DNS beállításait (a DNS kiszolgálón). A változások ezek után érvénybe lépnek az összes számítógépen, amelyek az adott kiszolgálóhoz szokták intézni DNS lekérdezéseiket. A DNS kiszolgálót kifejezetten a keresési sebesség maximalizálására optimalizálják; nagyobb adatbázis kezelésére is alkalmas, mint amekkorát az egyes számítógépek tudnának használni az izzadságos `hosts` állomány révén.

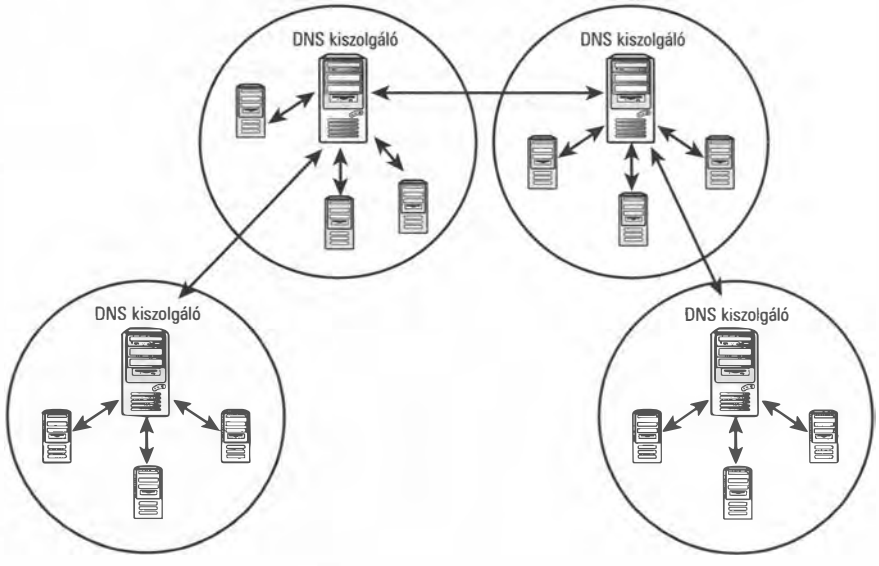


11.2 ábra
A DNS kiszolgáló biztosítja az egész hálózat számára a névfeloldást

A 11.2 ábrán látható bemutatja a DNS kiszolgáló előnyeit a `hosts` állománnyal megoldható névfeloldással szemben. A helyi hálózat egyetlen helyén oldható meg a konfiguráció, és lehetővé válik a hálózati erőforrások hatékonyabb kihasználása. A 11.2 ábrán bemutatott módszer azonban még nem ad választ arra a kérdésre, hogy mi a teendő egy nagyméretű, decentralizált hálózat esetén. A `hosts` állományhoz hasonlóan a 11.2 ábrán látható módszer sem terjeszthető ki olyan nagy hálózat ellátására, mint az internet. Egy ilyen névkiszolgáló nem tudna hatékonyan kezelni egy olyan adatbázist, amely az internet összes gépének címét tartalmazza. De ha ez még megoldható is lenne, az már semmiképp sem lenne kivitelezhető, hogy az adatbázist az egész internetre vonatkozóan egyetlen gépen tartsuk karban. Bárki is lenne érte a felelős, tudnia kellene arról, ha bárhol a világon az internetre kötnek egy gépet, vagy megváltoztatják a nevét.

Ennél még az is jobb, – érveltek a tervezők – ha minden irodában vagy intézményben beállítanak egy saját helyi névkiszolgálót (a 11.2 ábrának megfelelően), és lehetővé teszik, hogy ezek a névkiszolgálók kommunikáljanak egymással (11.3 ábra). Eszerint a forgatókönyv szerint a következő történik, amikor egy DNS ügyfél névfeloldási kérést küld a szervernek:

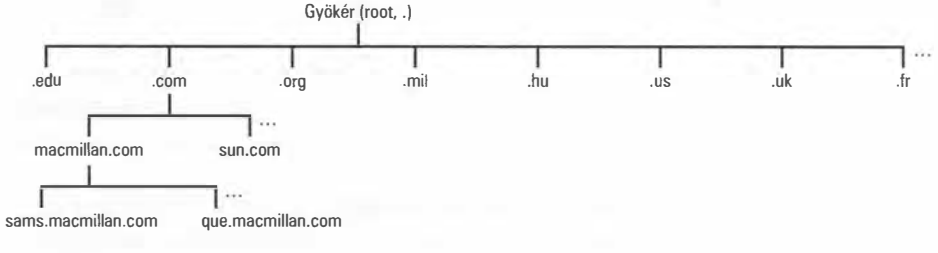
- Ha a névkiszolgáló megtalálja a keresett címet az adatbázisában, akkor azonnal visszaküldi azt az ügyfélnek.
- Ha azonban a névkiszolgáló nem találja meg a keresett címet, akkor kérést intéz más névkiszolgálókhoz, hogy keressék meg számára a kívánt címet; ha megkapta, akkor visszaküldi az ügyfélnek.



11.3 ábra

Nagy hálózatokban a DNS kiszolgálók egymással is kommunikálnak a névfeloldás biztosítása érdekében

Felmerülhet a kérdés, hogy a címkeresés megkezdésekor vajon honnan tudja az elsőként megszólított névkiszolgáló, hogy mely más névkiszolgálókat kell megszólítania. A keresési folyamat szoros kapcsolatban áll a DNS névtér szerkezetével. Ne felejtsük el, hogy a DNS nem pontosan olyan neveket használ, mint amilyenek az egyszerű gépnevek. Ahogy fejezetünk korábbi részében már említettük, a DNS teljes tartománynéveket (FQDN) használ. Egy teljes tartománynév tartalmazza a tartománynevet és a gépnevet is.



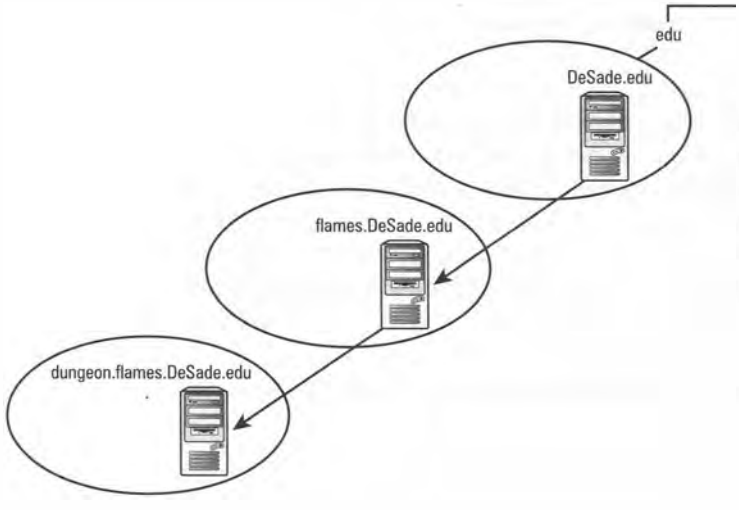
11.4 ábra

A DNS névtere

A DNS névtér tartományok többrétű összekapcsolásából alakul ki (lásd a 11.4 ábrát). Egy tartomány számítógépek bizonyos csoportját jelenti, amelyek a névtér egy adott részén (a szóban forgó tartománynéven) osztoznak. A DNS fastruktúra gyökerét egyetlen csomópont (a ***root***) képviseli. Ezt időnként „pontnak" (.) is hívják, bár a gyökérnek

ténylegesen az üres karakter felel meg. A („fejjel lefelé" ábrázolt) fastruktúra gyökere alatt helyezkedik el a **legfelső szintű tartománynevek** csoportja (**top level domains, TLDs**). A 11.4 ábra bemutatja a világ leghíresebb DNS névterének, az internetnek néhány legfelsőbb tartománynevét. A legfelső szintű tartománynevekhez tartozik a közismert `.com`, `.org` és `.edu`, valamint az egyes országokat azonosító tartománynevek, mint például a `.hu` (Magyarország), a `.us` (United States - Egyesült Államok), a `.uk` (United Kingdom - Egyesült Királyság), a `.fr` (Franciaország) és a `.jp` (Japán).

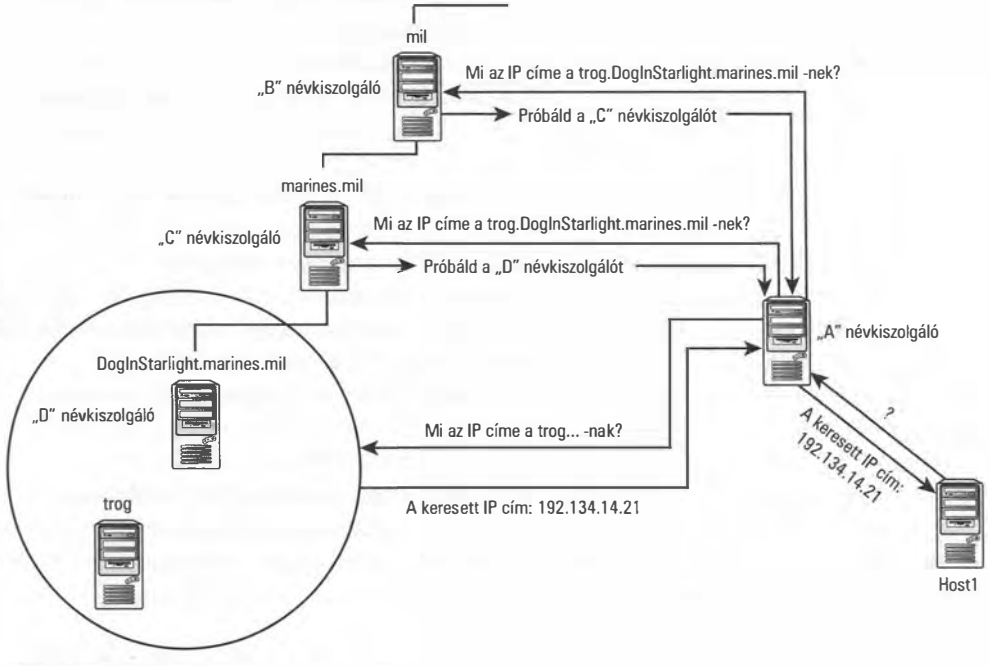
A legfelső szintű tartománynevek alatt helyezkedik el a következő réteg, amely (az internet esetében) vállalatok, intézmények, szervezetek neveivel kapcsolatos. A szervezeti nevek a legfelső szintű tartománynevek elé helyezendőek el, például a 11.5 ábrán látható DeSade Kollégium tartományneve `DeSade.edu`. A szervezet, mint a tartománynév tulajdonosa, létrehozhat további helyi alegységeket, altartományokat. Az ilyen helyi alegységek a szülő tartománynév elé illesztendőek. A DeSade Kollégium tűzijátékokkal foglalkozó alegységének neve például lehet `flames.DeSade.edu` (lásd a 11.5 ábrát), és a részleg népszerű „alvilági helye" lehet `dungeon.flames.DeSade.edu`. A DNS rendszer összesen 127 szintű tartománynevet tesz lehetővé (bár, valljuk meg, egy ilyen hosszú név már némi ellenszenvet váltana ki).



11.5 ábra
Egy elképzelhető DNS hierarchia

Aki sokat szörfözik az interneten, bizonyára megfigyelte, hogy a 11.5 ábrához hasonló szerkezetű kiterjesztett tartománynevek nem túl gyakoriak. A webhelyek, különösen a zsúfolt `.com` legfelső szintű tartománynév általában a `www` előtaggal jelennek meg, közvetlenül az intézmény nevével együtt: `www.ibm.com`. Tartsuk azonban szem előtt, hogy a weboldalak egyetlen (csoportnyi) kiszolgáló által is elláthatóak, amely(ek)hez nem kell „sok réteg". A többrétegű tartománynevek inkább ott kerülnek elő, ahol a nagy hálózati infrastruktúrát a hálózati adminisztrátorok kisebb egységekre bontják. A közigazgatási szektor tartománynevei (például a `.gov`) már gyakrabban osztódnak többrétegű nevekké.

A tartománynévből minden réteg neve kiderül, a fastruktúra csúcsától kezdődően. A `sams.com` tartományban működő névkiszolgáló minden olyan gép névfeloldási információját ismeri, amely ebben a (`sams.com`) tartományban található. Egy tartományért felelős névkiszolgáló továbbadhatja az altartományok névfeloldási teendőit egy másik kiszolgálónak. A `sams.com` tartomány felelős névkiszolgálója például átadhatja az `edit.sams.com` altartomány névfeloldási feladatkörét egy másik névkiszolgálónak. A feladat ellátásához az `edit.sams.com` altartomány név-cím adatait el kell juttatni az altartományért felelős névkiszolgálóra. A névfeloldási felelősség így görgethető tovább a fastruktúrában. Ily módon minden tartomány adminisztrátorának lehetősége van a nevek-címek egymáshoz rendelésére.



11.6 ábra
A névfeloldás folyamata

Amikor a hálózat valamely gépének szüksége van egy IP címre, akkor küld egy **rekurzív** lekérdezést a legközelebbi névkiszolgálónak. Az ilyen típusú lekérdezés arra utasítja a névkiszolgálót, hogy „adja meg a kért névhez tartozó IP címet, vagy mondja meg, hogy nem találja". Ha a névkiszolgáló nem találja a címet a rendelkezésére álló adatok között, akkor kezdeményez egy folyamatot, melyben más névkiszolgálókat hív segítségül a cím kiderítésére. Ez a folyamat látható a 11.6 ábrán. Az „A" névkiszolgáló **iteratív** (ismétlő) keresést hajt végre a cím kiderítésére. Ez a fajta lekérdezés azt mondja a legközelebbi névkiszolgálónak, hogy „adja meg a kért névhez tartozó IP címet, vagy mondja meg, hogy merre érdemes keresni". Összefoglalva: kliensgépünk egy egyszerű

rekurzív lekérdezést intéz a névkiszolgálóhoz, amely (nem találván a címet) ezután iteratív kereséseket küld más névkiszolgálóknak a név feloldása érdekében. Amikor megkapja a keresett címet, visszaküldi azt a kliensnek.

A DNS névfeloldás folyamat tehát a következő (figyeljük a 11.6 ábrát):

1. A `Host1` gép küld egy lekérdezést az „A" névkiszolgálónak, amelyben a trog.DogInStarlight.marines.mil gép IP címéről érdeklődik.
2. Az „A" névkiszolgáló leellenőrzi a saját adatbázisát, hogy megvan-e a kíván cím. Ha megvan, akkor visszaküldi a `Host1`-nek.
3. Ha az „A" névkiszolgáló nem találja a címet, akkor kezdeményezi a címkeresési folyamatot. Az „A" névkiszolgáló iteratív keresést küld a „B" kiszolgálónak (amely a .mil tartománynév legfőbb névkiszolgálója): mi a trog.DogInStarlight.marines.mil névhez tartozó cím?
4. A „B" névkiszolgáló nem tudja megválaszolni a kérdést, de el tudja küldeni az „A" névkiszolgálónak a „C" névkiszolgáló címét: ez a marines.mil tartomány névkiszolgálója.
5. Az „A" névkiszolgáló megkérdezi a címet a „C" névkiszolgálótól. A „C" névkiszolgáló nem tudja megválaszolni a kérdést, de el tudja küldeni neki a „D" névkiszolgáló címét: ez a DogInStarlight.marines.mil névkiszolgálója.
6. Az „A" névkiszolgáló elküldi az IP címre vonatkozó kérdését a „D" névkiszolgálónak .Ez megtalálja a trog.DogInStarlight.marines.mil gép címét és visszaküldi az „A" névkiszolgálónak, ami továbbküldi a `Host1` gépnek.
7. A `Host1` gép immár kapcsolatba tud lépni a trog.DogInStarlight.marines.mil géppel.

Ez a folyamat napjában több milliószor lejátszódik az interneten. Az alapjában véve egyszerű a forgatókönyvet időnként megbonyolítják a modern hálózatok további lehetőségeivel. Ilyen például a címtárazás, a DHCP és a dinamikus DNS. A legtöbb TCP/IP hálózat működése azonban a DNS névfeloldás fent vázolt formáján alapszik.

Fontos megjegyezni, hogy nem feltétlenül kell a faszerkezet minden csomópontjához külön névkiszolgálót üzembe állítani. Egyetlen névkiszolgáló több tartományt is kezelhet. Gyakran találkozunk azzal is, hogy több névkiszolgáló intézi egyetlen tartomány névfeloldási teendőit.

Egy tartomány bejegyzése (regisztrálása)

Az internet csak egy példa a DNS névtérre. Nem feltétlenül kell kapcsolatban lenni az internettel ahhoz, hogy DNS-t használjunk. Az internettől független hálózatokban nem érdekes, hogy milyen tartományneveket használunk. Ha azonban egy internettel összeköttetésben álló szervezet (mint amilyen például a BuddyCars.com) használni szeretné a saját tartományneveit, akkor regisztrálnia kell ezeket az erre illetékes szervnél.

Az ICANN (*Internet Corporation for Assigned Names and Numbers*) látja el a tartománynév regisztráció legfelső szintű teendőit, de bizonyos tartománynevek regisztrációját más csoportoknak adta át. Ezek közül sorolunk fel itt néhányat:

- .com, .org és .net – Számos cég (regisztrátor, angolul: *registrar*) jogosult arra, hogy a népszerű .com, .org és .net TLD-kre névkiszolgáló szolgáltatókat regisztráljon. Hasonló a helyzet a kevésbé gyakori .info, .museum, .name és .pro tartománynevekkel. A hivatalos, ICANN-től akkreditált regisztrátorok listáját megtaláljuk a *http://www.internic.net/regist.html* webhelyen.
- .gov – A .gov tartománynév az amerikai szövetségi kormányzat számára van lefoglalva. Az állami és a helyi (ön)kormányzatok az .us TLD-ből ágaznak el. A .gov tartománynév regisztrációs szolgáltatói a *http://www.registration.fed.gov* webhelyen találhatóak (újabban: *http://www.dotgov.gov* a *szerk.*).
- .mil – A .mil tartománynév az amerikai hadsereg számára van lefoglalva. A regisztrációval kapcsolatban a *http://www.nic.mil* webhelyen találhatóak információk.

Más tartománynevek számára (beleértve az országnevekhez rendelt tartományokat is) eltérő lehet a regisztrációs eljárás.

A névregisztrációért folyó küzdelem az utóbbi években egyre élesebbé vált. Bizonyos cégek azzal ügyeskednek, hogy egyes (értékesnek ítélt) tartományneveket előre regisztrálnak. Bizonyára járt már úgy az olvasó, hogy tévesen gépelt be egy címet a webböngészőbe, és a megjelenő weboldal azt a kérdést kezdte feszegetni, hogy szeretnénk-e regisztrálni a kívánt oldalt. Ha van egy cégnevünk, amit szeretnénk regisztrálni, akkor forduljunk közvetlenül egy hivatalos regisztrátorhoz. A szakemberek szerint nem érdemes úgy ellenőrizni egy név elérhetőségét, hogy közvetlenül beírjuk a böngészőbe. Voltak, akik azt tapasztalták, hogy a böngészőbe beírt nevet (varázslatos módon nem sokkal a keresés után) regisztrálta egy spekuláns. A nevesebb internetes cégek meg szokták tagadni, hogy kiszolgálják az ilyen taktikát űző spekulánsokat.

A DNS kezelése

Ha DNS szolgáltatást akarunk létrehozni hálózatukban, akkor ki kell választanunk (legalább) egy kiszolgálót, hogy legyen felelőse a tartománynév karbantartásának. Erre **elsődleges névkiszolgálóként** hivatkozunk a későbbiekben; a saját alhálózatára („zónáira") vonatkozó mindennemű információt helyi állományokból kell vennie. A tartományunkban eszközölt minden változást ezen a kiszolgálón kell karbantartani.

A legtöbb hálózaton van legalább még egy (biztonsági tartalékot jelentő) **másodlagos névkiszolgáló**. Ha elromlik az elsődleges névkiszolgáló, akkor ez a gép veszi át a szolgáltatást. A másodlagos névkiszolgáló az elsődleges névkiszolgáló zónaállományából veszi az információkat. Ezeknek az információknak az átkerülését **zónaátvitelnek** (*zone transfer*) hívjuk.

A névkiszolgálók harmadik típusát **csak-gyorstárazó** (*caching-only*) kiszolgálóknak hívjuk. A **gyorstár** a számítógép memóriájának egy része, amelyben a gyakran kért adatokat tároljuk. A csak-gyorstárazó kiszolgáló a helyi hálózat gépeinek névlekérdezéseit válaszolja meg. Más DNS kiszolgálókat is megkérdez a tartománynevekről és a (web vagy FTP szolgáltatást nyújtó) számítógépekről. A többi DNS kiszolgálótól kapott információkat eltárolja a gyorstárában, hogy vissza tudja adni, amikor ugyanennek az információnak a legközelebbi lekérdezése megérkezik.

Csak-gyorstárazó kiszolgálókat a helyi hálózatok számítógépei használnak a névfeloldáshoz. Az internet többi DNS kiszolgálói nem tudnak ezekről, és nem is intéznek hozzájuk lekérdezéseket. Akkor előnyös a használatuk, ha csökkenteni szeretnénk a többi névkiszolgálóra jutó terhelést. A csak-gyorstárazó kiszolgálókat könnyű karbantartani.

A DNS-t az erre kiszemelt névkiszolgálón, háttérprogramként futó szolgáltatásként érdemes megvalósítani. A Windows szervereknek van DNS szolgáltatása; egyes adminisztrátorok azonban jobban szeretnek harmadik féltől származó DNS megvalósítást futtatni. A Unix/Linux világban számos DNS megvalósítás létezik – a legnépszerűbb azonban kétségtelenül a *Berkeley Internet Name Domain* (BIND).

A DNS kiszolgálók beállításai

Zónának nevezzük a DNS-t használó gépeknek azon csoportját, amelyekhez azonos DNS szerverek vannak megadva. Egyszerűbb hálózatokon egy zónába tartozhat a teljes DNS tartomány. A *punyisp.com* tartomány például egyetlen zónának tekinthető a DNS beállítások tekintetében. Bonyolultabb hálózatokban előfordul, hogy egy-egy altartomány DNS beállításaira külön zónát jelölnek ki. A zónák kijelölése lehetővé teszi, hogy az adott alhálózat adminisztrátorai (akiknek közvetlen értesülései vannak az alhálózat jellemzőiről) közvetlenül beleszólhassanak alhálózatuk DNS beállításaiba. A *cocacola.com* tartomány DNS adminisztrátorai például a dallasi iroda DNS adminisztrátorainak fennhatósága alá rendelhetik a *dallas.cocacola.com* altartományt, hiszen nekik közvetlen rálátásuk van a *dallas.cocacola.com* gépeire.

Felmerülhet a kérdés, hogy mi a különbség a zóna és a tartomány között? Fontos tudni, hogy a finom szemantikai különbség mellett (vagyis, hogy a tartomány az a névtér része, a zóna pedig gépek csoportja) más eltérés is van a két fogalom között. Nem fedik egymást, mert az egyik hierarchikus, a másik mellérendelő. Fontoljuk meg az alábbiakat:

- Egy altartományba tartozó gép a szülő tartománynak is tagja. A *dallas.cocacola.com*-beli gép tagja a *cocacola.com*-nak is. Ezzel szemben, ha a *dallas.cocacola.com* zónát különválasztják, akkor a *dallas.cocacola.com*-beli gépek *nem* lesznek tagjai a cocacola.com zónának.
- Ha egy altartományt nem választanak külön, akkor nem fontos, hogy saját zónája legyen. Egyszerűen a szülő tartomány zónaállományával kezelhető.

A DNS zónák elkülönítésének módja függ a konkrét DNS kiszolgáló alkalmazástól. Egyelőre elég annyit megjegyezni, hogy a zóna azt jelenti, hogy a DNS kiszolgálók és gépek egy csoportját hasonlóképpen konfiguráljuk. A DNS adminisztrátorok (hatékonysági okokból) esetlegesen különálló zónákba választhatják le a névtér egy részét.

Zónaállományok

Ahogy az előző szakaszban megfogalmaztuk, a DNS zóna egy adminisztratív egység: számítógépek azon csoportját jelenti, amelyek a DNS névtér egy adott szeletébe tartoznak. A zóna DNS beállításait zónaállományban tároljuk. A DNS kiszolgálók a zónaállományban levő információkra támaszkodnak, amikor a hozzájuk intézett lekérdezéseket megválaszolják (vagy maguk kezdeményeznek lekérdezéseket). A zónaállomány egy adott szerkezetű szöveges állomány. Tartalma **erőforrás-bejegyzésekből** (*resource records*) áll. Egy erőforrás-bejegyzés egyetlen sorból áll, amely hasznos információkat ad meg a DNS kiszolgálónak. Álljon itt néhány jellegzetes erőforrás-bejegyzés típus:

- SOA — „*Start of Authority*" (Első számú hatóság). A SOA bejegyzés jelöli ki a zóna hitelesített névkiszolgálóját.
- NS — „*Name Server*" (Névkiszolgáló). Az NS bejegyzés névkiszolgálót jelöl ki a zónában. Egy zónának több névkiszolgálója (és így több NS bejegyzése) is lehet, de csak egyetlen SOA bejegyzés adható meg a hitelesített névkiszolgáló megnevezéséhez.
- A — Az „A" bejegyzés egy DNS névhez IP címet rendel
- PTR — A „PTR" (mutató) bejegyzés ennek fordítottja: IP címhez rendel DNS nevet.
- CNAME — „*Canonical NAME*" (Kanonikus (szabályos) név). Egy „A" bejegyzéssel megadott gépnévhez rendelhetünk egy álnevet a bejegyzéssel.

Ily módon a zónaállományból derül ki a DNS kiszolgáló számára:

- a zóna hitelesített névkiszolgálója
- a zóna (hitelesített és nem hitelesített) névkiszolgálói
- azon hozzárendelések, amelyekből a gépek DNS-neveiből › azok IP címe kiderül
- a zóna gépeinek álnevei (névváltozatai)

Más erőforrás-bejegyzés típusok olyasfélékhez adnak meg információkat, mint a levelezőszerverek (MX bejegyzések) vagy más közismert szolgáltatások (WKS, *Well-Known Services*). Egy zónafájl a következőképpen nézhet ki:

```
@ IN    SOA     boris.cocacola.com.   hostmaster.cocacola.com. (
        201.9       ; sorozatszám, amely a fájl megváltozásakor no
                    ;
        3600        ; frissítési ido (másodpercben)
        1800        ; várakozási ido (másodpercben)
        4000000     ; szavatossági ido (hétben)
        3600)       ; minimális élettartam
```

```
IN    NS    horace.cocacola.com.
IN    NS    boris.cocacola.com.
;
; gépnév -> IP cím hozzárendelések
;
localhost       IN   A    127.0.0.1
chuck           IN   A    181.21.23.4
amy             IN   A    181.21.23.5
darrah          IN   A    181.21.23.6
joe             IN   A    181.21.23.7
bill            IN   A    181.21.23.8
;
; Alnevek
;
ap              IN CNAME    amy
db              IN CNAME    darrah
bu              IN CNAME    bill
```

Figyeljük meg, hogy a SOA bejegyzésekben néhány olyan paraméter is megtalálható, amelyek (az elsődleges DNS kiszolgáló zónaadatainak mesterváltozata alapján) a másodlagos DNS kiszolgálók frissítési folyamatát szabályozzák. Szerepel az adatok között egy sorozatszám is, amely a magának a zónaállománynak a verziószámát jelenti. Ezen kívül az alábbi paramétereket láthatjuk:

- **Frissítési idő** – a másodlagos névkiszolgálók ennek megfelelő időközönként fordulhatnak az elsődleges névkiszolgálóhoz zónainformáció-frissítésért.
- **Várakozási idő** – a másodlagos névkiszolgálóknak legalább ennyi időt várakozniuk kell a sikertelen zónafrissítés esetén.
- **Szavatossági idő** – az az időintervallum, ameddig a másodlagos névkiszolgálóknak meg kell őrizniük frissítetlen bejegyzéseiket.
- **Minimális élettartam** – az elküldött zónabejegyzések minimális érvényességi ideje.

A SOA bejegyzés jobboldali kifejezése valójában a zónáért felelős személy email címe, csak az első pontot ki kell cserélni egy @ jelre.

Ez a példa természetesen egy végsőkig leegyszerűsített zónaállományt mutat be. Vannak sokkal nagyobb fájlok is, bejegyzések százaival, és más, kevésbé gyakori bejegyzéstípusokkal, amelyek különféle beállítási finomításokat tesznek lehetővé. A zónaállomány neve (és esetenként formátuma is) változhat a DNS kiszolgáló szoftvernek megfelelően. Példánkban a népszerű BIND (*Berkeley Internet Name Domain*) névkiszolgáló szintaxisát használtuk. Ez a névkiszolgáló a leggyakoribb az interneten.

Meg kell említenem, hogy a szöveges konfigurációs állományok szerkesztésének becses gyakorlata napjainkban egyre kevésbé divatos. Több DNS kiszolgáló alkalmazáshoz tartozik grafikus felhasználói felület, amely a zavaró részleteket elrejti a felhasználó elől. A dinamikus DNS (amely fejezetünkben részletesen is előkerül) még egy réteget húz ki a beállítási részletek fölé.

A fordított keresés zónaállománya

A DNS névfeloldáshoz szükség van még egy sajátos állományra, ez pedig a „fordított keresés zónaállománya". Ha egy ügyfél megad egy IP címet, és a hozzá tartozó nevet kéri, akkor erre az állományra van szükség. Az IP címek baloldali része „általános", a jobboldali részen vannak a „jellemző" adatok. A tartománynevekben éppen fordítva van: a baloldali rész a „speciális", és a jobboldali (`.hu` vagy `.com`) az „általános". A fordított keresés zónaállományában fordítva állnak a hálózati cím bájtjai, hogy így könnyen megfeleltethető legyen egymásnak az „általános" és a „speciális" rész. A `192.59.66.0` hálózat megnevezéséhez a `66.59.192.in-addr.arpa` adandó meg.

Ebben az állományban minden bejegyzés olyan, hogy a gép azonosítója után ott áll az `in-addr.arpa` rész – ezzel rövidítik az „inverse address" (fordított cím) kifejezést. Az `arpa` egy legfelső szintű tartománynév; az internet ősét jelentő `ARPAnet` névből maradt fenn.

Az A és B osztályú hálózatokban rövidebb a fordított keresési zónaállomány, hiszen kevesebb hálózati bit megadására van szükség. Egy `43.0.0.0` című A osztályú hálózatban a fordított keresési zónaállománynak tartalmaznia kell a `43.in-addr.arpa` címet; a `172.58.0.0` című B osztályú hálózatban pedig a `58.172.in-addr.arpa` címet kell tartalmaznia.

DNS segédprogramok

Hálózatunk megfelelő névfeloldási működésének kipróbálásához bármilyen hálózati segédprogramot használhatunk, amely névfeloldásra támaszkodik. Egy webböngésző, egy FTP ügyfélprogram, egy telnet vagy egy ping segédprogram is meg tudja mondani, hogy számítógépünk sikeresen megbirkózik-e a névfeloldás feladatával. Ha az IP cím megadásával csatlakozni tudunk egy erőforráshoz, de gépnév vagy az FQDN megadásával nem, akkor nagy valószínűséggel a névfeloldással van baj.

Ha számítógépünk `hosts` állományt és DNS-t is használ, akkor ne felejtsük el ideiglenesen kikapcsolni vagy átnevezni a `hosts` állományt, amíg a DNS-t vizsgáljuk. Máskülönben nem deríthető ki könnyen, hogy a névfeloldás a `hosts` állomány vagy a DNS révén történt-e meg. A következő részben vázoljuk, hogy hogyan lehet a `ping` segítségével megvizsgálni a DNS működését. Egy következő szakaszban pedig az `nslookup` segédprogramról lesz szó, amely számos DNS beállítási és hibakeresési lehetőséget is nyújt.

A névfeloldás ellenőrzése a ping segítségével

A ping egyszerű és hasznos segédprogram, jól használható a DNS beállítások ellenőrzésére. A ping küld egy jelzést egy másik számítógépnek, és várja a választ, a „visszhangot" (innen a neve). Ha a válasz megérkezik, akkor tudható, hogy az összeköttetés sikerült. Ha tudjuk a másik gép IP címét, akkor könnyen megpingelhető a kiszemelt gép:

```
ping 198.1.14.2
```

Ha megérkezik a válasz, akkor tudható, hogy a két számítógép között lehetséges az összeköttetés kiépítése.

Ilyenkor meg lehet próbálni a DNS név segítségével történő válaszkérést:

```
ping kiszemeltgep.tavolihalozat.hu
```

Ha IP cím alapján érkezik válasz a kiszemelt géptől, de DNS név alapján nem, akkor névfeloldási probléma van. Ha DNS név alapján is megjön a válasz, akkor jól működik a névfeloldás.

A ping-ről a 14. órában még bővebben is lesz szó.

A névfeloldás ellenőrzése az nslookup segítségével

Az nslookup segédprogram lehetővé teszi, hogy kérdéseket intézzünk a DNS kiszolgálókhoz, és ránézzünk az egyes erőforrás-bejegyzésekre. Igen hasznos a DNS hibák keresésekor. Az nslookup segédprogram két üzemmódban futhat:

- kötegelt (batch) módban – ilyenkor az nslookup elindításakor paraméterként adhatjuk meg a bemenő adatokat. A program elvégzi a paraméterekkel jelzett műveleteket, kijelzi az eredményt, majd leáll.
- interaktív módban – ebben az üzemmódban paraméterek nélkül indíthatjuk el az nslookup segédprogramot, amely ekkor bekéri a szükséges adatokat. Ezután elvégzi a kért műveleteket, kijelzi az eredményt, majd újra visszaadja a parancssort (a következő paraméterlistára várva). Többnyire ezt az üzemmódot szeretik a rendszergazdák, mert kényelmesebb akkor, amikor több műveletet szeretnénk végrehajtatni.

Az nslookup számos kapcsolóval és paraméterrel futtatható. Az alábbiakban megnézzük a leggyakrabban használtakat, hogy legyen némi fogalmunk arról, hogy hogyan is működik a program.

Az nslookup interaktív módban történő futtatásához írjuk be egy parancssori felületről:

```
nslookup
```

A 11.7 ábrán láthatjuk a program egy lehetséges futását. A válaszok mindig annak a DNS kiszolgálónak a nevével és IP címével kezdődnek, amelyiket az `nslookup` éppen használ, például:

```
Default Server: dnsserver.Lastingimpressions.com
Address: 192.59.66.200
>
```


11.7 ábra
Az nslookup futása

A kacsacsőr (>) jelzi az `nslookup` promptját (parancssori bemenetét).

Az `nslookup`-nak 15-féle kapcsolója van – ezek mindegyike befolyásolja a program futását. A leggyakoribbak a következők:

- `?` vagy `help` — megjeleníti az `nslookup` kapcsolóit és opcióit.
- `server` — megadhatjuk, hogy mely DNS szervert szeretnénk használni a lekérdezéseinkkel.
- `ls` — kilistázza egy tartomány gépneveit, ahogy az a 11.7 ábra közepén is látható.
- `ls -a` — (*aliases*) kilistázza egy tartomány alapértelmezett (kanonikus) gépneveit és ezek álneveit, ahogy az a 11.7 ábrán is látható.
- `ls -d` — (*domain*) kilistázza egy tartomány erőforrás-bejegyzéseit, ahogy ahogy az a 11.7 ábra alján is látható.
- `set all` — kilistáz minden beállítási értéket

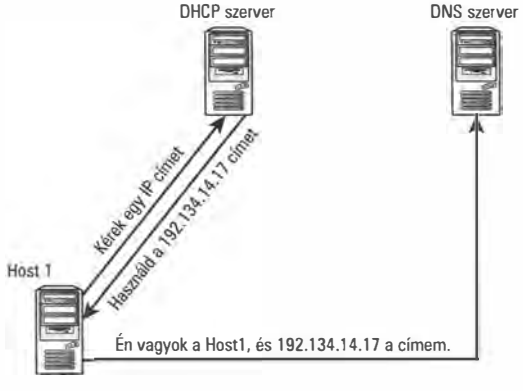
Az `nslookup`-pal nemcsak a saját DNS szervereinket vizsgálhatjuk meg, hanem elvileg bármely DNS szervert. Ha internetszolgáltatóhoz (ISP-hez) kapcsolódunk, akkor legalább két DNS névkiszolgáló IP címével rendelkeznünk kell. Az `nslookup` IP címet

vagy tartománynevet is tud használni. Az `nslookup`-ot a `server` paranccsal állíthatjuk át másik DNS kiszolgáló használatára (IP cím vagy FQDN megadásával). Tegyük fel, hogy 192.203.230.10 egy elsődleges szerver IP címe; ekkor a `server 192.203.230.10` paranccsal tudunk hozzá kapcsolódni az `nslookup` parancssorából. Ezután elvileg bármilyen tartománynevet megadhatunk, akár azt is, hogy samspublishing.com. Megnézhetjük azokat az IP címeket, amelyek ehhez a tartománynévhez vannak regisztrálva. A valóságban azonban a legtöbb kereskedelmi DNS kiszolgáló (és elsődleges kiszolgáló) vissza szokta utasítani az `ls` parancs végrehajtását, mivel a kimenet jelentős forgalmat generál, és biztonsági kérdéseket is felvet ezeknek az információknak a közreadása.

Dinamikus DNS

A DNS-ről az eddigiekben olyan képünk alakulhatott ki, mint amit valamilyen (lényegében) állandó „gépnév és IP cím összekapcsolási rendszerhez" terveztek. A mai hálózatokban (ahogy az a következőkben kiderül) az IP címet gyakran dinamikusan kapják meg a számítógépek. Más szóval: minden alkalommal, amikor elindul egy számítógép, a DHCP-n (*Dynamic Host Configuration Protocol*) keresztül kap egy új IP címet. Ez azt is jelenti, hogy ha a gépet regisztrálni kell a DNS-be (és a neve már ismert), akkor a DNS kiszolgálónak valami módon tudomást kell szereznie a gépnek kiosztott IP címről.

A dinamikus IP cím kiosztás gyors elterjedéséhez alkalmazkodniuk kellett a DNS kiszolgálók tervezőinek. A legtöbb megvalósítás (köztük a BIND is) lehetővé teszi a DNS bejegyzések dinamikus frissítését. A 11.8. ábrán is bemutatott tipikus forgatókönyv szerint a DHCP kiszolgáló kioszt egy IP címet az ügyfélgépnek, majd pedig frissíti a DNS kiszolgálót: átadja neki az ügyfélgép új címét. A DHCP-ről bővebben is fogunk tanulni a 12. órában.



11.8 ábra
Dinamikus DNS-frissítés

NetBIOS névfeloldás

A NetBIOS-t eredetileg az IBM fejlesztette ki: ez egy névfeloldási rendszer és egyben API (alkalmazásprogramozási felület). A Microsoft Windows hálózatokban használják. A NetBIOS nevet magához a (Windows-t futtató) számítógéphez társítják. A NetBIOS számítógépnévvel azonosítja a gépet az Intéző (Explorer) és a Sajátgép (My Computer). A NetBIOS nem a TCP/IP hálózathoz lett kifejlesztve. A NetBIOS számítógépnév egy kissé redundáns a TCP/IP hálózatban, mert igazából ugyanolyan szerepet játszik, mint a gépnév (*hostname*). A Microsoft igyekezett nem túlhangsúlyozni (sőt, visszafogta) a NetBIOS szerepét a Windows 2000/XP-ben, de a Windows Vista újra visszatért a NetBIOS-központúsághoz. A Windows Vista teljes mértékben támogatja a NetBIOS névfeloldási módszert; ily módon manapság már tömegesen üzemelnek olyan számítógépek, amelyek működésre a NetBIOS-ra támaszkodik. Nem lenne tehát teljes a névfeloldásról szóló fejezetünk a NetBIOS áttekintése nélkül.

A legutóbbi Windows verziókban a felhasználó nézőpontjából elmosódik a különbség a NetBIOS és DNS névfeloldás között. A Windows igazából párhuzamosan karbantartja a két rendszert. Az ismerős Windows-os számítógépnév (a beállításoktól függően) betöltheti a DNS gépnév és a NetBIOS gépnév szerepét is.

Mivel a NetBIOS közvetlen üzenetszórással (*broadcasts*) működik, egy kisebb hálózatban dolgozó felhasználónak semmi teendője nincs a NetBIOS névfeloldás beállítását illetően. Nagyobb hálózatban azonban már bonyolultabb a NetBIOS működtetése. Ilyenkor üzembe állítanak ún. WINS szervereket: ezek a névkiszolgálók oldják fel a NetBIOS neveket IP címekké. A DNS-hez hasonlóan megadhatunk statikus `LMHosts` állományokat is, amellyel lehetővé válik a helyi névfeloldás. A következő szakaszokban közelebbről is megvizsgáljuk a NetBIOS névfeloldást.

NetBIOS névfeloldási módszerek

A TCP/IP hálózatokban a NetBIOS névfeloldás végső célja megegyezik a DNS névfeloldáséval: a NetBIOS névhez meg kell találni az aktuális IP címet.

A NetBIOS név 15 karakteres lehet: ilyesféle szokott lenni, mint `Workstation1`, `HRServer`, `CorpServer...` A NetBIOS névnek egyedinek kell lennie a hálózatban.

Igazából 16 karakternyi a NetBIOS név, csak a 16. karaktert az alább fekvő alkalmazási réteg használja – a felhasználónak általában nincs lehetősége arra, hogy közvetlenül megadja. A karakterek szerepére még visszatérünk ezen az órán.

A NetBIOS nevek (akárcsak az egyszerű gépnevek) „egyrétű névteret" alkotnak; nincs lehetőség hierarchikus vagy más névtér-struktúra kialakítására. Az alábbi szakaszokban megvizsgálunk néhány lehetőséget arra, hogy hogyan szokás a NetBIOS neveket IP címekké alakítani.

- Az üzenetszórásos névfeloldás
- `LMHosts` állomány révén történő névfeloldás
- WINS névfeloldás

Üzenetszórásos névfeloldás

A névfeloldás egyik lehetősége az üzenetszórásos megoldás. Akkor beszélünk **üzenetszórásról**, ha egy számítógép a saját alhálózatába tartozó összes többi számítógépnek üzenet küld. A névfeloldáskor azt kérdezi a szóban forgó számítógép, hogy ráismeri-e valamelyik gép a keresett névre. Az összes számítógép veszi az adást, de csak az válaszol, amelyik ráismer saját nevére: ez visszaküldi az IP címét.

A névfeloldásnak ezt a módját B-csomópontos (*B-node*; ahol a *B* betű a *Broadcast*-ra utal) névfeloldásnak is hívják. Ez jól működik helyi hálózatban. Hátránya, hogy az információ nem tud átjutni az útválasztókon, mert ezek alapértelmezetten blokkolják a szórt üzeneteket.

Az üzenetszórás erős hálózati terhelést tud okozni azzal, hogy minden gépet „megzavar", és ez hátrányosan érintheti a hálózat működését. Az útválasztót emiatt nem adják tovább a szórt üzeneteket.

Az üzenetszórásos névfeloldás egyszerű, és használata nem igényel különösebb beállításokat. A hálózati kártya és a TCP/IP hálózati program telepítése (a Windows operációs rendszeren) már elegendő ahhoz, hogy a NetBIOS üzenetszórásos névfeloldási módszerével meg lehessen keresni más számítógépeket.

LMHosts állományok révén történő névfeloldás

A Windows rendszerek `LMHosts` állományok révén is fel tudják oldani az IP címeket. Az `LMHosts` állománynak hasonló a szerepe, mint korábban tárgyalt hosts állománynak. Az `LMHosts` állomány NetBIOS nevekhez társít IP címeket. Az IP címek a baloldalon állnak, a hozzájuk tartozó számítógépnevek pedig a jobboldalon, majd # jel után beilleszthető egy megjegyzés, amely utalhat például a szóban forgó gép szerepére. Az `LMHosts` állomány használata statikus IP címzést feltételez, melyben minden IP cím egy előre rögzített NetBIOS névhez tartozik. Minden számítógéphez tartozik egy külön

LMHosts állomány, amelyet kézileg lehet beállítani. Ha egy új számítógép kerül a hálózatba, akkor a többi (LMHosts-t használó) gép mindaddig nem fogja őt látni, amíg kézileg be nem írják az új nevet az egyes gépek LMHosts fájljába.

Ha egy hálózat egyetlen szegmensből áll, akkor nem érdemes LMHosts állományt használni, mivel egy adott hálózatban a NetBIOS névfeloldás üzenetszórással is kiválóan működik. Bár esetenként hasznos lehet teljesítménybeli vagy (régi, üzenetszórásra képtelen rendszerekkel való) kompatibilitási megfontolásokból az LMHosts állomány használata az ilyen hálózatokban is. A több alhálózatból álló nagyobb hálózatokban az üzenetszórásos névfeloldás csak az útválasztóig használható. Ilyen esetben a számítógépeknek LMHosts vagy WINS alapú NetBIOS névfeloldást kell használniuk (ez utóbbiról hamarosan szó lesz). Az LMHosts arra is felhasználható, hogy a tartományvezérlőt (*domain controller*) egy másik hálózati szegmensből vegyük. (A tartományvezérlő a tartományalapú Windows környezetben történő hitelesítéshez kell.)

Az LMHosts névben az LM a Microsoft LAN Manager (*Microsorft helyi hálózatkezelő*) nevére utal – a Windows NT előtt ez volt a hálózat kezeléséért felelős segédprogram.

Egy egyszerűbb LMHosts fájl így nézhet ki:

```
192.59.66.205    marketserv      #file server for marketing department -
                                  a marketing részleg fájlszervere
192.59.66.206    marketapp       #application server for marketing -
                                  marketinges alkalmazásszerver
192.59.66.207    bobscomputer    #bob's workstation - bob munkaállomása
```

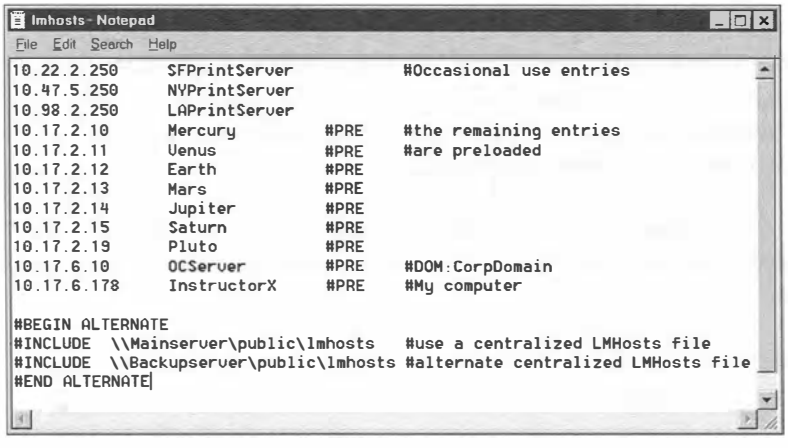
A legutóbb használt NetBIOS nevek a NetBIOS névgyorstárban őrződnek. Amikor egy felhasználó megpróbál azonosítani egy számítógépet, akkor a rendszer először mindig a NetBIOS névgyorstárban néz körül, hogy nincs-e ott a keresett név. Ezután kerül sor az LMHosts állomány bejegyzéseinek vizsgálatára. Nagyobb LMHosts állomány esetén ez időigényes lehet, így a keresések gyorsításához megtehetjük, hogy (a #PRE kulcsszó megadásával: 11.9 ábra) egyes gyakran használt bejegyzéseket már a gép indulásakor beolvastatunk a gyorstárba. Az LMHosts állomány teljes egészében beolvasásra kerül, így akkor lesznek a későbbi keresések hatékonyak, ha a #PRE kulcsszóval ellátott bejegyzések az állomány végén helyezkednek el. Mivel ezeket csak egyszer célszerű beolvasni, az LMHosts fájl végére téve őket csökkenthetjük a felesleges újraolvasás esélyét.

Az nbtstat segédprogram használatával megnézhetjük és meg is változtathatjuk a NetBIOS névgyorstár tartalmát. A parancssorból kiadott nbtstat -c utasítással tekinthetjük meg a gyorstár aktuális tartalmát.

A hosts vagy az LMHosts statikus állományok használata kényelmetlen, mivel ezeket minden gépen külön kell karbantartani. A központosítás kérdését részben megoldhatjuk úgy, hogy az LMHosts állományban az #INCLUDE paranccsal olyan állományt olvasta-

tunk be, amely egy előre kijelölt gépen található. Ezzel a kulcsszóval egy szerverről emelhetjük be a szükséges bejegyzéseket az egyes helyi gépek `LMHosts` állományába. A változtatásokat tehát csak egy helyen (a szerveren) kell eszközölni, és ezek minden felhasználó gépén hozzáférhetőek lesznek.

Ha egynél több `#INCLUDE` bejegyzésünk van, akkor a `#BEGIN ALTERNATE` és az `#END ALTERNATE` utasítások közé kell őket írni, ahogy az a 11.9 ábrán is látható.



11.9 ábra
Egy LMHosts állomány tartalma

Ahogy korábban már említettük, az `LMHosts` állományt arra is felhasználhatjuk, hogy a Windows tartalomvezérlőt egy másik hálózati szegmensből vegyük. Ezt a `#DOM` kulcsszóval tehetjók meg: ez olyan `LMHosts` bejegyzést jelöl, amely tartományvezérlőt reprezentál.

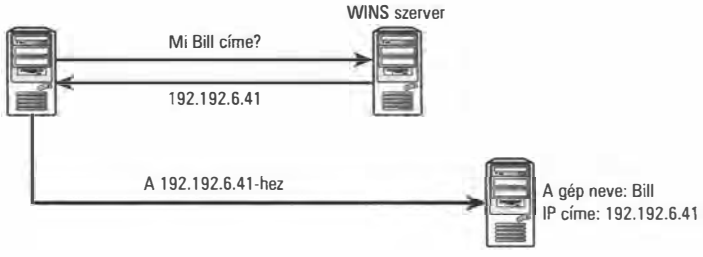
WINS segítségével történő névfeloldás

A Windows Internet Name Service (WINS) azért jött létre, hogy ugyanazokat a problémákat orvosolja az `LMHosts` esetében, mint ami a `hosts` állományok kapcsán felmerült, és amelyet a DNS kialakítása oldott meg. Ha egy ügyfélgépnek egy másik gép IP címére van szüksége, akkor a WINS szerverhez fordulhat információért.

A WINS elnevezés valójában a Microsoft-féle *Netbios Names Server* (NBNS) implementációhoz kapcsolódik. A NetBIOS névkiszolgálók az 1001-es és 1002-es RFC-ben vannak leírva.

A WINS a regisztrált NetBIOS nevek adatbázisát tartja karban. Ezek különféle objektumok lehetnek: felhasználók, munkacsoportok, számítógépek és az ezeken futó szolgáltatások. Az adatbázis azonban nem kézileg szerkesztett szövegfájlból alakul ki, hanem az egyes gépek (elinduláskor) dinamikusan regisztrálják nevüket és IP címüket a WINS szervernél.

A WINS szerver NetBIOS névfeloldási kéréseket fogad és küld (11.10 ábra). A 11.10 ábrán látható WINS szerver kísértetiesen emlékeztet a 11.2 ábrán látható DNS kiszolgálóra. A WINS szerver ugyanazt teszi a NetBIOS névfeloldással kapcsolatban, amit a DNS névkiszolgáló a tartománynév-feloldással kapcsolatban. Az egyrétű NetBIOS névtér azonban nem teszi lehetővé a DNS által biztosított hierarchikus rendszerű névfeloldási megoldásokat.



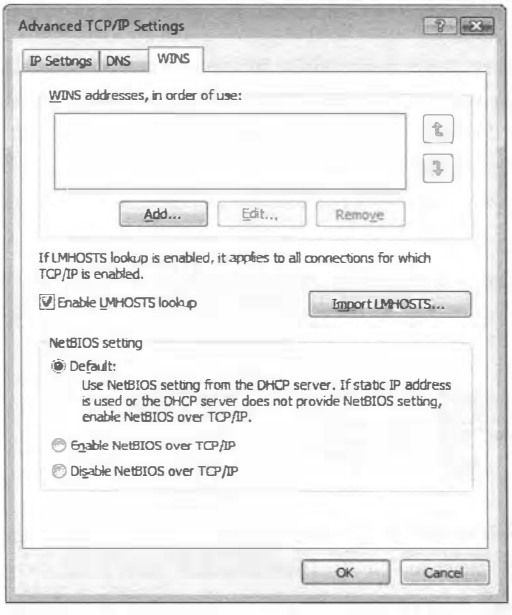
11.10 ábra
WINS NetBIOS névfeloldás

A Windowsban többféle módon is beállíthatóak az ügyfélgépek a WINS használatára. Ha a számítógép DHCP-től dinamikusan kapja a TCP/IP beállításokat (lást a 12. órát), akkor a WINS beállítása is történhet DHCP-n keresztül. Kézileg is beírhatóak a WINS kiszolgáló címei. A TCP/IP beállítási párbeszédpaneljén keresztül más, NetBIOS névfeloldáshoz kapcsolódó információk is megadhatóak.

A WINS beállításának konkrét lépései Windows-verzió-függőek. Windows Vistán a (11.11 ábrán is látható) „Haladó TCP/IP beállítások" párbeszédpanel WINS fülén lehet megadni a WINS beállításokat. A „Haladó TCP/IP beállítások" párbeszédpanel eléréséhez kövessük az alábbi lépéseket:

1. Válasszuk a Start menüben a Hálózat pontot.
2. Menjünk a Hálózatmegosztási Központba (*Network Sharing Center*).
3. Válasszuk a Hálózati csatlakozások kezelését (*Manage Network Connections*).
4. Jobb kattintás a beállítandó hálózati csatlakozásra, majd a Tulajdonságok (*Properties*) kiválasztása következik – ehhez azonban rendszergazda jogosultsággal kell rendelkeznünk.
5. Válasszuk az Internet Protokoll 4. verziót (TCP/IPv4), majd a Tulajdonságokat (*Properties*).
6. A TCP/IPv4 Tulajdonságok párbeszédpanelen kattintsunk a „Haladó" (*Advanced*) gombra.
7. Válasszuk a WINS fület.

Ahogy az a 11.11 ábrán is látható, a WINS fül lehetővé teszi, hogy a WINS kiszolgálók címeit kézileg adjuk meg. Bekapcsolhatjuk az `LMHosts` keresést is, és beemelhetünk egy meglévő `LMHosts` állományt. Megjegyzendő, hogy a rendszer alapértelmezetten a DHCP kiszolgálótól várja a NetBIOS beállításokat, de az is megadható, hogy a DHCP helyett „NetBIOS a TCP/IP-n" legyen érvényben.



11.11 ábra
A WINS beállítása Windows Vistán

Ha egy (WINS használatára beállított) ügyfélgépet bekapcsolnak, akkor a következő lépések zajlanak le:

1. **A szolgáltatás elindítása** – Ahogy a számítógép betölt, különféle szolgáltatások indulnak el, melyek közül néhányat más számítógépekkel is tudatni kell.
2. **Regisztrációs kérés** – Hogy a többi számítógép is tudomást szerezzen a szolgáltatásról, regisztrálni kell azt. A WINS ügyfélgép összecsomagolja a NetBIOS nevet és a számítógép IP címét egy névregisztrációs kérésbe, és elküldi a WINS kiszolgálónak. Amikor ez megérkezik, a WINS kiszolgáló ellenőrzi az adatbázisban, hogy nincs-e már regisztrálva az adott név.
   Ha nincs, akkor felveszi az adatbázisba a név+IP cím adatpárt, és visszajelzést ad az ügyfélgépnek a sikeres regisztrációról. Ha már létezik a kért név, a WINS adatbázisban, akkor a WINS kiszolgáló küld egy ellenőrzést annak a gépnek, amelyikkel ütközik a keresett név. Ha ez a gép válaszol, akkor negatív visszajelzést kap a névregisztrációt megkísérlő ügyfélgép. Ha az eredetileg regisztrált gép nem válaszol, akkor a WINS lehetővé teszi a regisztrációt, felülírva a korábbi, ütközőnek látszó adatpárt.
3. **Bérlés** – Ha számítógépünk sikeresen regisztrálta NetBIOS nevét és szolgáltatásait a WINS kiszolgálónál, akkor ezek a nevek foglalttá („béreltté"; *leased*) válnak. Ez azt jelenti, hogy a számítógép egy ideig (például 6 napig) jogosan használhatja az adott NetBIOS nevet. Természetesen az ügyfélgép megújíthatja a regisztrációt még a lejárta előtt. Ez általában a lejárati idő felénél (példánkban 3 nap után) meg szokott történni.

Említettük korábban, hogy a NetBIOS név 16. karaktere nem a felhasználó által állítható be. A WINS regisztráció folyamata során a 16. karaktert a WINS kiszolgáló fűzi a névhez. Ez attól függően jön létre, hogy milyen szolgáltatásról van szó. A számítógépnevek, munkacsoport-nevek, szolgáltatások sokasága esetén nem ritka, hogy egyetlen számítógépről is 5-10 WINS adatbázis-bejegyzés készül.

A WINS névfeloldási folyamat másik példájaként képzeljük el, hogy egy felhasználó a „Hálózati környezet" (*Network Neighborhood*) alkalmazás segítségével keres egy másik számítógépet a hálózaton, amelyhez csatlakozni szeretne. Ilyenkor egy névkeresési kérést állít össze a számítógép, melyben jelzi, hogy NetBIOS névkeresésről van szó – és elküldi a WINS szervernek. A WINS szerver fogadja a lekérdezést, és megkeresi adatbázisában az illeszkedő nevet. Ha talál megfelelőt, akkor egy válaszcsomagban visszaküldi a hozzá tartozó IP címet. Miután az ügyfélgépnek rendelkezésére áll a keresett számítógép IP címe, közvetlenül tud már vele kommunikálni.

A NetBIOS névfeloldás vizsgálata

A NetBIOS névfeloldás működését megvizsgálhatjuk alkalmas segédprogramokkal. Az egyik leghasznosabb eszköz a `net view` parancs, amely lehetővé teszi, hogy feltérképezzük egy kiszolgálóhoz tartozó megosztott könyvtárakat. (Emléketetőül: az **megosztott könyvtár** (*share point*) egy olyan könyvtár, amelyhez más számítógépek csatlakozhatnak, és az ott levő állományokat, erőforrásokat megnézhetik, kezelhetik.) Ehhez a vizsgálathoz válasszunk egy olyan gépet, amelynek több megosztott könyvtára van. A parancssorból gépeljük be (behelyettesítve a keresett számítógépnevet):

```
net view \\szamitogepnev
```

Ha a `net view` parancs sikeresen elvégzi a névfeloldást, és rendelkezésünkre áll az IP cím, akkor a program megjeleníti a megosztott könyvtárak listáját.

A NetBIOS névfeloldás ellenőrzésére használhatjuk az örökzöld `ping` segédprogramot is. A legtöbb Windows rendszeren, amelyen megfelelően működik a NetBIOS névfeloldás, képesek vagyunk NetBIOS név révén is „megpingelni" (visszhangadásra kérni) más számítógépeket. Ha egy számítógépnek `Judit` a neve, akkor működnie kell a

```
ping Judit
```

parancsnak – és meg kell érkeznie a visszhangnak.

Összefoglalás.

A névfeloldás lehetővé teszi, hogy IP címek helyett „beszédes", megjegyezhető számítógépneveket használjunk. Ezen az órán először az egyszerű gépnév alapú, majd a DNS alapú névfeloldásról volt szó. Tanultunk a NetBIOS névfeloldási rendszerről is, amelyet Microsoft hálózatokban használnak.

Kérdések és válaszok

K *Mi az a tartománynév?*

V A tartománynév az a név, amellyel egy hálózat azonosítható. A tartománynevet központi hatóság regisztrálja, hogy biztosított legyen a név egyedisége.

K *Mi az a gépnév (hostname)?*

V A gépnév egy egyszerű név, amellyel egy számítógépet megnevezhetünk, és amelyhez adott IP cím tartozik.

K *Mi az az FQDN?*

V Az FQDN (*Fully Qualified Domain Name*) vagy teljes tartománynév a gépnév és a megfelelő tartománynév (ponttal történő) összekapcsolásával jön létre. Ha a gépnév `bigserver`, a tartománynév pedig `mycompany.com`, akkor a teljes tartománynév `bigserver.mycompany.com`.

K *Mik azok a DNS erőforrás-bejegyzések?*

V Az erőforrás-bejegyzések a DNS zónaállományában található bejegyzések. A különféle erőforrás-bejegyzések különböző számítógépeket vagy szolgáltatásokat azonosítanak.

K *Milyen erőforrás-bejegyzés típus használatos az álnevek megadására?*

V `CNAME`; az ezzel megadott gépnévhez álnevet rendelhetünk egy „A" bejegyzésben.

K *Hogyan lehet központosítva kezelni az LMHosts állomány bejegyzéseit?*

V Központosított `LMHosts`-kezelés valósítható meg az `#INCLUDE` beemelő-parancs segítségével. Az így kezdett sorban megadható egy távoli szerveren található `LMHosts` állomány, amelyet egyszerű karbantartani.

K *Hogyan lehet elérni, hogy a NetBIOS névgyorstárban benne legyenek az általunk megadott NetBIOS bejegyzések?*

V A gyorstárazni kívánt bejegyzéseket `#PRE` kulcsszóval kell ellátni az `LMHosts` állományban.

Gyakorlatok

- Számítógépünk parancssorából adjuk ki a `ping localhost` parancsot, és írjuk le a kapott IP címet.
- Számítógépünk parancssorából adjuk ki a `hostname` parancsot, és írjuk le a kapott gépnevet.
- Adjunk ki egy `ping` parancsot saját gépünk nevével.
- Ha gépünkhöz tartozik tartománynév, adjunk ki egy `ping` parancsot az FQDN nevünkre is.
- Ha hálózatunkon van beállítva DNS névkiszolgáló, akkor adjuk ki az alábbi `ping` parancsokat:

```
ping www.internic.net
ping www.whitehouse.gov
```

- Az `nslookup` parancs segítségével keressük meg internetszolgáltatónk DNS névkiszolgálóit.

Kulcsfogalmak

Tekintsük át röviden a frissen tanult kulcsfogalmakat:

- **DNS** (*Domain Name System*) – A TCP/IP hálózatok erőforrásait megnevező rendszer.
- **Tartománynév** (*Domain name*) – A DNS névtér egy adott hierarchikus szintjéhez tartozó név.
- **Teljes tartománynév** (*Fully Qualified Domain Name ; FQDN*) – A gépnév és a megfelelő tartománynév összekapcsolásával jön létre.
- **Gépnév** (*hostname*) – A gépnév egy egyszerű név, amellyel egy adott számítógépet (host) megnevezhetünk.
- **`LMHosts`** – Az az állomány, amelyben IP címeket társíthatunk NetBIOS nevekhez.
- **Erőforrás-bejegyzés** – A zónaállomány egy adott bejegyzése. Számos erőforrás-bejegyzés típus van, mindegyiknek más a rendeltetése.
- **WINS** (*Windows Internet Naming Service*) – A WINS szerver a NetBIOS névkiszolgálók Microsoft-féle implementációja.
- **Zónaállomány** – A DNS kiszolgálók által használt szöveges konfigurációs állomány.

12. ÓRA

A beállítások automatizálása

Ebben az órában a következőkről lesz szó:

- Dinamikus címkiosztás
- DHCP
- Hálózati címfordítás (NAT)
- Konfigurációmentes hálózathasználat (*Zeroconf*)

A régi szép időkben minden ügyfélgép egyetlen statikus IP címmel rendelkezett, amelyet egy erre kijelölt konfigurációs állományban adott meg a rendszergazda. Ennek a megváltoztatásához a rendszergazda kézi beavatkozására volt szükség. A mai hálózatkezelés azonban már rugalmasabb és kényelmesebb megoldásokat igényel. Miért ne lehetne a TCP/IP beállításokat hálózati szolgáltatásokkal automatizálni? Ezen az órán pontosan erről lesz szó: a TCP/IP címkiosztás automatizálási módszereiről.

Az óra anyagának elsajátítása után az olvasó képes lesz

- vázolni a DHCP-t és az általa biztosított előnyöket
- vázolni az IP címek DHCP-n keresztül történő kiosztását (bérlését)
- vázolni a hálózati címfordítás célját
- bemutatni, hogy hogyan használhatóak a „konfigurációmentes" (*Zeroconf*) protokollok

Miért van szükség egy kiszolgáló által kiosztott IP címekre?

Ahogy az előző órán tanultuk, a TCP/IP hálózat használatához rendelkeznie kell a számítógépeknek egy IP címmel. Az IP címzés rendszerét eredetileg úgy tervezték, hogy már eleve minden számítógéphez legyen beállítva egy adott IP cím. Ezt hívják **statikus IP címzésnek**. Már a bekapcsolástól kezdve ismeri minden számítógép a saját IP címét, és azonnal képes használni a hálózatot. A statikus IP címzés jól működik kis méretű, állandó hálózatokon, de a nagy hálózatokban, ahol gyakran történnek változások (új gépek jönnek, vagy meglévők távoznak a hálózatból) a statikus IP címzés korlátai már jelentős problémákat okoznak. A statikus IP címzés főbb hátrányai a következők:

- Több beállításra van szükség – Minden ügyfélgépet egyedileg kell beállítani. Az IP címtér (vagy más paraméter, például a DNS névkiszolgáló címének a) megváltozásakor minden egyes ügyfélgépet egyedileg át kell állítani.
- Több címre van szükség – Minden számítógéphez saját IP címet kell fenntartani, akár be van kapcsolva, akár nem.
- Rugalmatlanabb – Ha más alhálózathoz szeretne csatlakozni egy számítógép, akkor kézileg át kell állítani.

Ezen korlátozások kiküszöbölésére egy alternatív IP címzési rendszert fejlesztettek ki, amelyben az IP címeket külön kérésre kapják meg az egyes gépek, mégpedig a DHCP protokollon keresztül. A DHCP elődje a BOOTP volt, amelyet merevlemez nélküli számítógépek elindításához használtak. Az ilyen számítógép a teljes operációs rendszert a hálózaton keresztül tölti be a bekapcsolás után. A DHCP egyre népszerűbb lett az utóbbi években, mert egyre több a nagyméretű, dinamikusan változó hálózat, és egyre szűkösebben állnak rendelkezésre kioszthatő IP címek.

Manapság már az internet-eléréssel rendelkező számítógépek többsége DHCP-n keresztül kapja meg a beállításait. Az a kisméretű útválasztó/tűzfal, amely révén az internet elérkezik az olvasó otthonába, valószínűleg egyúttal DHCP kiszolgálóként is üzemel.

Mi az a DHCP?

A DHCP protokoll arra használható, hogy egy TCP/IP hálózatban egy számítógép automatikusan megkapja a hálózati paramétereit. A DHCP szabvány az 1531-es RFC-ben van leírva. Más RFC-k is érintik a témát, mint az 1534-es, 1541-es, 1231-es és 2132-es – ezek a címzésmód megjavítását vagy gyártóspecifikus DHCP-megvalósításokat adnak meg. A DHCP kiszolgáló számos TCP/IP beállítást átadhat a DHCP ügyfélnek – természetesen magát az IP címet, de ezen kívül még az alhálózati maszkot és a DNS névkiszolgálók címeit is.

Mivel a DHCP kiszolgáló osztja ki az IP címet, csak a DHCP kiszolgálót kell statikus IP címmel ellátni. Az ügyfélgépnek egyedül azt az információt kell megadni, hogy ő bizony DHCP-n keresztül fog IP címet (és más hálózati paramétereket) kapni. A TCP/IP beállítás többi részletét a kiszolgáló intézi el. Ha a TCP/IP hálózat valamely jellemzője megváltozik, akkor a hálózati adminisztrátornak elég a DHCP kiszolgálót frissítenie – nem kell minden egyes ügyfélgépen kézileg helyesbítenie az adatokat.

Minden ügyfélgép véges időre kapja meg az IP címet. Ha a bérleti idő lejártakor már nem használja, akkor ez a cím bármelyik gépnek kiosztható. A DHCP által kiosztott IP címek rendszerének ebből ered az egyik fő előnye: a hálózaton nincs szükség annyi IP címre, mint ahány gép van összesen.

A DHCP különösen is fontos a mai környezetekben. Számos dolgozó hozza-viszi a kézi számítógépét, nagyobb vállalat esetén akár egyik irodából a másikba. Ha egy ilyen munkára használt kézi számítógép statikus IP címmel rendelkezne, akkor mindannyiszor manuálisan át kellene állítani, valahányszor új alhálózatba lép be a dolgozó. Ha viszont úgy van beállítva a gép, hogy DHCP-n keresztül kapja meg az IP címét, akkor minden TCP/IP beállítást automatikusan megkap, mihelyt a felhasználó rácsatlakozik egy alhálózatra.

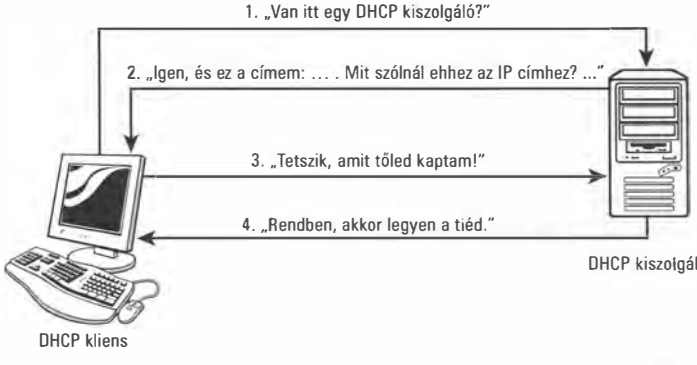
Hogyan működik a DHCP?

Amikor egy DHCP ügyfélgép elindul, a TCP/IP-t kezelő program bekerül a memóriájába, és dolgozni kezd. Mivel kezdetben a TCP/IP protokollverem nem kapott IP címet, nem képes irányított adatcsomagokat kapni vagy küldeni. Képes azonban üzenetszórásos (*broadcast*) adatküldésre és -fogadásra. Ez az alapja a DHCP működésének. Az IP címek DHCP kiszolgálótól történő bérlésének négylépéses folyamata a következő (lásd a 12.1 ábrát is):

1. **DHCPDISCOVER** – (*Felkutatás*) A DHCP ügyfél azzal kezdi a folyamatot, hogy szórt üzenetet küld a 68-as UDP kapura, amelyet kifejezetten a BOOTP és DHCP kiszolgálók használnak. Az első adatcsomagot DHCP felkutatási (*discover*) üzenetnek hívjuk – ezzel az ügyfélgép jelzi, hogy szeretné felvenni a kapcsolatot egy DHCP kiszolgálóval, amely jó esetben az meg is kapja az adatcsomagot. Ebben több információ is szerepel, a legfontosabb azonban a DHCP ügyfélgép fizikai címe.
2. **DHCPOFFER** – (*Ajánlat*) A DHCP kiszolgáló (amelyet úgy állítottak be, hogy az adott hálózaton egy adott intervallumból osszon ki IP címeket) válaszként egy DHCP ajánlat (*offer*) adatcsomagot küld vissza szórt üzenetként annak a számítógépnek, amely kiadta a DHCP felkutatási kérést. Ez a szórt üzenet (amely az ügyfélgép fizikai címét is tartalmazza) a 67-es UDP kapura küldi a kiszolgáló. Az adatcsomagban szerepel még néhány fontos információ: a DHCP kiszolgáló fizikai és IP címe, valamint az ügyfélgépnek felajánlott IP cím és alhálózati maszk. Eddig a pontig még megteheti a DHCP ügyfél, hogy több DHCP kiszolgálótól is elfogad-

jon ajánlatot, hiszen nyugodtan elképzelhető, hogy több DHCP kiszolgáló is működik a közelben, és mindegyik kínál egy ajánlatot. A legtöbbször egyszerűen az első érkező ajánlattal foglalkozik az ügyfélgép, a többivel nem.

3. DHCPREQUEST – (*Igénylés*) Miután az ügyfél kiválasztotta az ajánlatot, összeállít és elküld egy DHCP igénylési (*request*) adatcsomagot. A DHCP igénylési adatcsomag tartalmazza a megcélzott DHCP kiszolgáló (amelynek az ajánlatát elfogadta az ügyfél) IP címét, valamint a DHCP ügyfél saját fizikai címét is. A DHCP igénylésnek kettős célja van. Egyrészt jelzi a kiválasztott DHCP kiszolgálónak, hogy az ügyfél kéri a felajánlott IP cím (és más paraméterek) regisztrálását, másrészt pedig az összes többi DHCP kiszolgáló számára is informatív az adatcsomag: megtudhatják belőle, hogy az ő ajánlatukat elutasította az ügyfélgép.
4. DHCPACK – (*Nyugtázás*) Amikor a kiválasztott DHCP kiszolgáló (amelyiktől a kedvezményezett ajánlat érkezett) megkapja az igénylést jelző adatcsomagot, összeállít egy adatcsomagot, amely lezárja a bérlési folyamatot. Ezt hívjuk DHCP nyugtázásnak (*ack*, *acknowledgment*). A DHCP nyugtázási csomag tartalmazza a DHCP ügyfél IP címét és alhálózati maszkját – opcionálisan szerepelhet az adatok között az alapértelmezett átjáró, néhány DNS névkiszolgáló, és esetleg egy vagy több WINS kiszolgáló is. Az IP címeken kívül más információk is elküldhetőek az ügyfélgépnek, például a NetBIOS csomóponttípus, amely megváltoztathatja a NetBIOS névfeloldás sorrendiségét. Van még három kulcsmező a DHCP nyugtázó üzenetben – mindegyik időintervallumokat jelent. Az egyik jelzi a bérbeadási időszakot. A másik kettő (`T1` és `T2`) akkor használatos, amikor az ügyfélgép meg akarja hosszabbítani a kapott paraméterek használati időtartamát.



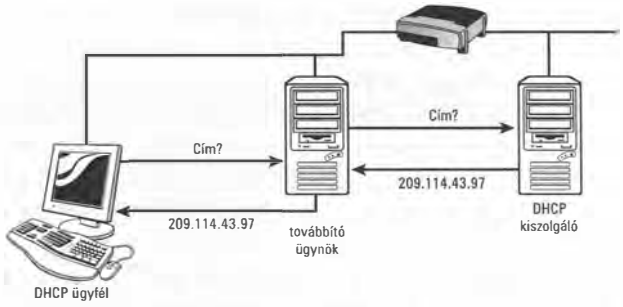
12.1 ábra
A DHCP kiszolgáló IP címet ad egy kliensgépnek

Továbbító (relay) ügynökök

Ha a DHCP ügyfél és szerver ugyanazon az alhálózaton működik, akkor a folyamat pontosan úgy zajlik le, ahogy az imént vázoltuk. Ha azonban a két szereplőt útválasztók választják el egymástól, akkor a folyamat némileg bonyolódik. Az útválasztók általában nem továbbítanak szórt üzeneteket más alhálózatoknak. A DHCP működtetéséhez

ilyenkor egy közvetítőre van szükség. Ez tetszőleges gép lehet a DHCP ügyfél hálózatában, de általában maga az útválasztó játssza ezt a szerepet. A közvetítőt mindkét esetben „**BOOTP továbbító ügynöknek**" vagy „**DHCP továbbító ügynöknek**" hívják.

A továbbító ügynök fix IP címmel rendelkezik, és ismeri a DHCP kiszolgáló IP címét is. Ily módon (a közbeeső útválasztóktól függetlenül) mindig képes a DHCP kiszolgálónak adatcsomagokat küldeni és ezeket fogadni. Mivel a továbbító ügynök és a DHCP ügyfél ugyanazon az alhálózaton van, kényelmesen kommunikálhatnak egymással akár üzenetszórással is (12.2 ábra).



12.2 ábra

A továbbító ügynök teszi lehetővé a DHCP ügyfél számára, hogy elérje a másik alhálózatban levő DHCP kiszolgálót.

A továbbító ügynökök a 68-as UDP kapura küldött szórt üzeneteket figyelik. Ha ilyen DHCP kérést észlelnek, továbbküldik a DHCP kiszolgálónak. Ha pedig választ kapnak a DHCP kiszolgálótól, akkor erről szórt üzenetben adnak hírt a helyi alhálózat gépeinek.

Az itt vázolt sematikus kép kissé elnagyolt, és kimaradt néhány apró részlet, de a közvetítő ügynökök funkciójának a lényegét tartalmazza.

Az utóbbi években elterjedt az a gyakorlat, hogy a DHCP kiszolgáló szerepét is az útválasztó lássa el. Ily módon a legtöbb hálózatban feleslegessé vált a DHCP továbbító ügynökök feladatköre is. További részleteket is megtudhatnak a továbbító ügynökökről az 1542-es RFC-ből.

Nem minden útválasztó képes BOOTP/DHCP továbbító ügyfél szolgáltatások ellátására. Az erre a feladatkörre is alkalmas útválasztókat RFC 1542-kompatibilis útválasztóknak hívják.

A DHCP időintervallum-mezői

A DHCP ügyfelek egy megadott időszakra kapják meg az IP címet a DHCP kiszolgálótól. A tényleges bérleti időt általában be lehet állítani a DHCP kiszolgálón. A nyugtázó (*ack*) üzenetben küldi el a DHCP szerver a `T1` és `T2` értékeket, amelyeknek a bérleti idő meghosszabbításakor van szerepe. A `T1` időintervallum adja meg, hogy mikor kell megkezdeni a meghosszabbításra vonatkozó kérést. Ez általában a bérleti idő fele. A példa kedvéért tegyük föl, hogy a teljes bérleti idő nyolc nap.

Amikor már négy napja használja az adott gép a kapott paramétereket (az IP címet és egyebeket), akkor küld egy DHCP kérést az IP cím megújítására vonatkozóan annak a DHCP kiszolgálónak, amelyiktől a bérleti jogot kapta. Ha a DHCP kiszolgáló online állapotban van, akkor a megújítási kérést általában egy DHCP nyugtázó (*ack*) visszajelzés követi. A korábban vázolt négylépéses folyamattal ellentétben most nem szórt üzenetként történik a kérés és nyugtázás küldése, hiszen minden résztvevő tisztában van a másik aktuális IP címével: közvetlenül egymásnak küldik tehát az adatcsomagokat.

Ha a DHCP kiszolgáló nem elérhető a megújítási kérés pillanatában (ami példánkban a bérleti idő 50%-a, azaz négy nap), akkor vár egy darabig az ügyfélgép. Amikor elért a bérleti idő 75%-ához (példánkban ez hat nap), akkor újra megkísérli a meghosszabbításra vonatkozó kérést. Ha ez is sikertelen, akkor az időszak 87,5%-ánál, (azaz példánkban a hetedik napon) is történik egy kérés. Eddig a pontig a DHCP ügyfél ahhoz a DHCP kiszolgálóhoz intézi kéréseit, amelyiktől a bérleti jogot kapta. Ha az időszak 87,5%-ánál is sikertelen a hosszabbítási kérés, akkor lép életbe a `T2` időintervallum-paraméter. Ezt követően az ügyfélgép már bármelyik DHCP kiszolgálóhoz intézheti kéréseit. Ha a DHCP ügyfél képtelen meghosszabbítani aktuális IP címét, és új cím kérése is sikertelen, akkor a bérleti idő lejártakor abba kell hagynia a nála lévő IP cím használatát – ezzel elesik a TCP/IP hálózat normál használati lehetőségétől is.

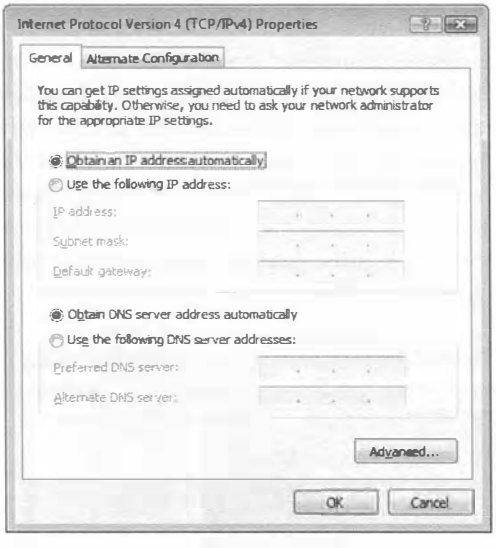
A DHCP ügyfélgépek beállítása

A DHCP ügyfél egy köteg beállítási információt kap a DHCP kiszolgálótól. Ebben az IP címen kívül szerepelnek másféle beállítási adatok is. Mivel az ügyfél szinte mindent a DHCP kiszolgálótól kap, maga az ügyfél szinte alig igényel beállításokat. A DHCP használata a legtöbbször alapértelmezetten be van kapcsolva. Ha ebben nem vagyunk biztosak, vessünk egy pillantást a TCP/IP beállítási párbeszédpanel megfelelő jelölőmezőjére.

Egy Windows Vista gépet az alábbi módon lehet beállítani DHCP használatára:

1. A Start menüből nyissuk meg a Vezérlőpanelt.
2. Kattintsunk duplán a Hálózatokra.
3. Kattintsunk a Hálózati kapcsolatok kezelésére.

4. Kattintsunk jobb egérgombbal a Helyi hálózati kapcsolatokra, majd válasszuk a Tulajdonságokat.
5. Válasszuk az „Internet protokoll, 4. verzió"-t (TCP/IPv4), majd kattintsunk a Tulajdonságokra.
6. A TCP/IPv4 Tulajdonságok párbeszédpanelben válasszuk az „IP cím automatikus hozzárendelésé"-t (12.3 ábra). Figyeljük meg, hogy a párbeszédpanel arra is lehetőséget ad, hogy a DNS névkiszolgáló címét is automatikusan kapja meg a gépünk. A DHCP kiszolgáló az IP címen kívül még számos más hálózati paramétert is felkínál.
7. Kattintsunk OK-ra a TCP/IPv4 párbeszédpanelben, majd ugyancsak OK-ra a Helyi hálózati kapcsolatoknál a Tulajdonságok ablakon.



12.3 ábra
Windows Vista ügyfélgép beállítása DHCP használatára

A DHCP kiszolgáló beállítása

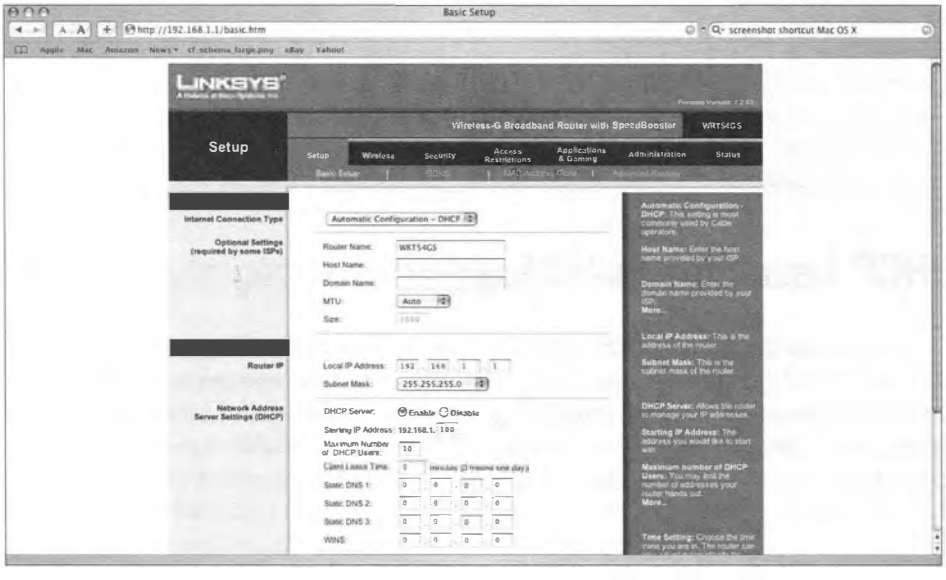
A legtöbb felhasználónak nincs szüksége arra, hogy tisztában legyen egy DHCP kiszolgáló beállításával. Ha pedig egy közép- vagy nagyvállalat rendszeradminisztrátora az olvasó, akkor valószínűleg ennél a könyvnél részletesebb dokumentáció is a keze ügyében van, amelynek segítségével aprólékosan be tudja állítani a kiszemelt számítógépet, hogy DHCP kiszolgálóként üzemeljen. A Windows grafikus felhasználói felületet is rendelkezésre bocsát (DHCP Kezelő) a DHCP kiszolgáló beállításához.

A Linux rendszerek a `dhcpd`-n, vagyis a DHCP daemon-on (háttérprogramon) keresztül valósítják meg a DHCP szolgáltatásokat. A `dhcpd` telepítésére vonatkozó utasítások az operációs rendszer közreadójától függően változhatnak. A DHCP beállításai a `/etc/dhcpd.conf` szöveges állományban találhatóak.

A `/etc/dhcpd.conf` állomány tartalmazza mindazokat az IP cím beállítási adatokat, amelyeket a DHCP háttérprogramnak tudnia kell ahhoz, hogy az ügyfélgépeknek (megfelelő időszakra) alkalmas IP címeket tudjon kiosztani. A `/etc/dhcpd.conf` állomány egyéb beállítási adatokat is tartalmaz, mint például az üzenetszórási cím, a tartománynév, a DNS kiszolgáló címe és az útválasztók címei. Álljon itt egy példa a `/etc/dhcpd.conf` állományra:

```
default-lease-time 600;
max-lease-time 7200;
option domain-name "macmillan.com";
option subnet-mask 255.255.255.0;
option broadcast-address 185.142.13.255;
subnet 185.142.13.0 netmask 255.255.255.0 {
range 185.142.13.10 185.142.13.50;
range 185.142.13.100 185.142.13.200;
}
```

Ahogy fejezetünkben már említettük, a DHCP szolgáltatást gyakran ugyanaz a hálózati hardvereszköz végzi, amely az útválasztó/tűzfal is egyben. Érdemes elolvasni az otthoni útválasztónkhoz adott kézikönyvet – abban minden valószínűség szerint szerepel a DHCP beállításának módja. Az útválasztó eszközöket a legtöbbször webes felületen tudjuk beállítani (12.4 ábra). Jelentkezzünk be az útválasztónk beállítási weboldalára, és próbáljuk megváltoztatni a DHCP beállításokat. A legtöbb esetben természetesen nincs szükség az alapértelmezett DHCP paraméterek megváltoztatására.



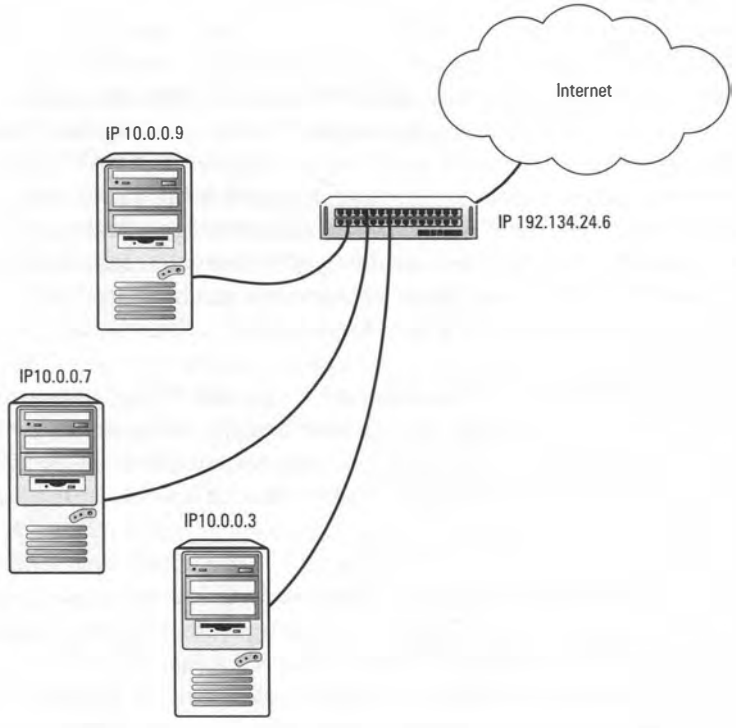
12.4 ábra

DHCP beállítása egy otthoni útválasztó eszközön

Vannak esetek, amikor egy-egy eszközhöz célszerű fix IP címet rendelni akkor is, amikor a hálózat többi tagja dinamikusan kap IP címet. Ilyen lehet például egy hálózati nyomtató, amelynek címét nem szeretnénk újra meg újra hirdetni a többi gépnek. Egyes útválasztók lehetővé tesznek **IP cím lefoglalást** (IP reservation), amely révén lehetővé válik, hogy egyes fizikai (MAC) címekhez konkrét IP címet társítsunk. Így az adott gépeknek állandó IP címe lesz.

Hálózati címfordítás (NAT)

A szakemberek felfedezték, hogy ha a DHCP kiszolgáló látja el IP címmel az ügyfeleket, akkor semmi akadálya annak, hogy ezek az IP címek „házi használatra" készüljenek – nem szükséges, hogy az egész világon egyedi, „hivatalos" internetcímeket osszunk ki a belső hálózaton. Ha magának az útválasztónak „hivatalos" internetcíme van, akkor proxyként működhet a hálózat ügyfélgépei számára – kéréseket fogadhat az ügyfelektől és átalakíthatja ezeket a kéréseket úgy, hogy azok illeszkedjenek az internet névtér követelményeihez. Ma már számos útválasztó és DHCP eszköz ellát hálózati címfordítási (*Network Address Translation*, NAT) szolgáltatást is.



12.5 ábra

Hálózati címfordítást (NAT) végző eszköz.

A NAT eszközök elfedik a helyi hálózat paramétereit, sőt, még azt a tényt is, hogy az adott gép mögött egy helyi hálózat húzódik. A 12.5 ábrán láthatunk egy NAT eszközt. A NAT eszköz átjáróként szolgál a helyi hálózat gépei számára az internet felé. A NAT eszköz mögötti helyi hálózat tetszőleges névteret használhat. Amikor egy helyi gép megpróbál kapcsolódni egy internetes erőforráshoz, helyette a NAT eszköz végzi el a kapcsolódást. Az internetes erőforrástól érkező adatcsomagok aztán (a NAT eszköz által karbantartott címtáblázatnak megfelelően) átalakulnak helyi címet tartalmazó adatcsomagokká, és így jutnak el a kapcsolatot kezdeményező helyi géphez.

A NAT eszközök biztonsági szempontból is hasznosak, mert gátolják a külső támadót annak megismerésében, hogy milyen is a belső hálózat. A külvilág számára a NAT eszköz (és a mögötte levő hálózat) egyetlen gépnek látszik, amely az internetre csatlakozik. Ha a támadó tudja is a helyi hálózaton működő kiszemelt számítógép címét, nem tud kapcsolatot nyitni a helyi hálózatra, mivel a helyi címek rendszere nem illeszkedik az internetes névtérhez. A 4. órában említettük, hogy vannak IP cím tartományok, amelyek „magánhálózatok" számára vannak lefoglalva:

```
10.0.0.0     - 10.255.255.255
169.254.0.0 - 169.254.255.255
172.16.0.0   - 172.31.255.255
192.168.0.0 - 192.168.255.255
```

A NAT eszközök általában ezekből a tartományokból jelölnek ki IP címeket. Ezek a címek nem használhatóak útválasztásra a hagyományos módon, így az *egyedüli* mód a NAT kliensgépek elérésére a címfordítási folyamaton keresztül vezet. A NAT révén csökkenthető a szervezet számára lefoglalandó internet-kompatibilis IP címek száma. Valójában csak a NAT eszközként szolgáló útválasztónak van szüksége internet-kompatibilis címre. A „hivatalos" internetcímek gazdaságos felhasználási lehetősége és a magánhálózatok megnövelt biztonsága különleges népszerűséget hozott a NAT eszközöknek az elmúlt években, mind az otthoni, mint a testületi hálózatokban.

Egy biztonsági rendszer természetesen nem mindig olyan megingathatatlan, mint amilyennek először tűnik. Még az olyan bolondbiztos rendszer is megtörhető, mint a NAT. Előfordul, hogy a NAT eszközökhöz lehetővé tesznek az internet irányából is használható adminisztrátori felületet, és megfelelő elővigyázatosság nélkül ez komoly biztonsági rést jelenthet.

A NAT elterjedése a támadási módszerek átalakulásához vezetett, amely a magánhálózatok természetes védelmét vette célba. A támadók gyakran úgy jutnak be egy magánhálózatba, hogy ráveszik az ügyfeleket, hogy „hívják meg őket". A mai behatolók gyakran jól célzott hivatkozásokat küldenek szét nem létező weboldalakra, és így veszik rá a felhasználókat, hogy kapcsolatot kezdeményezzenek romboló szerverrendszerekhez. Az ilyenfajta támadások miatt nem tanácsos a kéretlen email üzenetek hivatkozásaira rákattintani. A modern webböngészők némelyike el tudja fogni az efféle támadásokat, amelyek Cross Site Scripting-en (XSS) vagy más webes támadási módszeren alapulnak.

Konfigurációmentes hálózat

Elgondolkozhatunk azon, hogy mi történne akkor, ha az összes hálózati ügyfélgép DHCP használatára lenne beállítva, és a DHCP kiszolgáló nem működne. Az ügyfélgépek egy része életképes maradna és várna a kommunikáció lehetőségére, de nem lenne statikus IP címe, sem pedig lehetősége arra, hogy DHCP-n keresztül szerezzen egyet. Lehet olyan eset is (bár manapság ez már egyre ritkább), hogy egy felhasználó olyan PC-kből szeretne egy munkacsoportot kialakítani, amelyeknek nincs szüksége internetelérésre, sem pedig egy speciális DHCP/útválasztó eszköz elérésére.

Néhány operációs rendszer gyártó olyan technológiát fejlesztett ki, amely lehetővé teszi, hogy egy helyi hálózat számítógépei úgy lépjenek egymással kapcsolatba, hogy ahhoz sem statikus beállításokra, sem pedig DHCP általi beállításokra ne legyen szükség. Néhány régebbi LAN protokoll, mint például a Windows-on használatos NetBEUI vagy az Apple hálózatokon használt AppleTalk lehetővé tette ezt a „nemhivatalos" beállításmentes csatlakozási lehetőséget, és a gyártók olyan megoldást kerestek, amely erre a TCP/IP keretein belül is lehetőséget ad.

Ennek a folyamatnak az első lépéseként kialakult egy módszer, amelyet „adott alhálózaton egyedi címzésnek" („**Link Local Addressing**", Ipv4LL) hívnak. Az „adott alhálózaton egyedi címzés" része volt az Apple rendszereknek az OS 9-től kezdve, és a Windowsba is bekerült a Windows 98 óta.

Az Ipv4LL Windows-os változatát a Microsoft „automatikus privát IP címzésnek" (**Automatic Private IP Addressing; APIPA**) hívja. Ha egy Windows-os számítógépnek nincs statikus IP címe és nem tud dinamikusan sem szerezni, akkor kitalál magának egyet a privát (útválasztásnál használhatatlan) címtartományból, a 169.254.0.0-169.254.255.255 intervallumból. Ha a helyi hálózat más számítógépei is hasonló helyzetbe kerülnek, akkor ők is keresnek maguknak egy (még nem használt) számot ebből a tartományból. Ezek a számítógépek már sikerrel kommunikálhatnak egymással a helyi hálózaton. Mivel ezek a címek nem használhatóak útválasztáskor, az említett számítógépek nem érik el az internetet (és a helyi hálózaton kívül eső erőforrásokat).

Az APIPA lényege, hogy nem igényel beállítást – így a beállításokról nem szükséges bővebben beszélnünk. A legtöbb Windows változat rendszerleíró adatbázisa (*registry*) tartalmaz egy kulcsot az APIPA használatának a ***kikapcsolására***. Az ezzel kapcsolatos részletekhez nézzük meg a Windows dokumentációját.

Az APIPA használata esetenként okozhat bonyodalmat. Ha például a többi hálózati gép normál módon van beállítva, és van egy gép, amelyik nem akarja az igazságot (elérhetetlen), akkor érdemes megnézni, hogy ez a gép nem veszítette-e el látóteréből a DHCP kiszolgálót, és nem adott-e magának önkényesen egy APIPA címet, amely összeférhetetlen a helyi címek névterével.

A **Zeroconf** nevű újabb technológia sokkal hatékonyabb és teljesebb konfigurációmentes megoldást kínál, mint az eddig említettek. A Zeroconf kiterjeszti az Ipv4LL filozófiáját, és szinte teljes hálózati környezetet tud kialakítani kis helyi hálózatokon. A Zeroconf rendszert az Apple Macintosh rendszereken **Bonjour** néven valósították meg. A legújabb Windows rendszerekben is megtalálható egy ehhez hasonló konfigurációmentes technológia, amely némileg eltérő protokollokat használ. A Linux/Unix rendszerek konfigurációmentes Zeroconf megvalósítása az Apple változathoz hasonló Avahi.

Ennek az új konfigurációmentes környezetnek három fontos összetevője van:

- Adott alhálózaton egyedi címzés (*Link Local Addressing*) – A számítógépek kijelölhetnek maguknak egy címet az IPvLL 169.254.0.0-169.254.255.255 privát címtartományából (a részleteket lásd az előző bekezdésekben)
- Többes-küldéses (*Multicast*) DNS – DNS névfeloldási rendszer, amely nem igényel kiszolgálót, sem pedig előre beállított `hosts` állományokat. Az egymáshoz tartozó nevek és IP címek összerendelése speciális IP címre és kapuszámra küldött kérésekkel valósul meg. A többi gép figyeli az erre kijelölt címre küldött kéréseket, és megfelelő információkat juttat vissza a válaszban.
- DNS szolgáltatás-felderítés (*DNS Service Discovery*) – Lehetőséget biztosítunk az ügyfélgépek számára, hogy felderítsék a hálózaton elérhető szolgáltatásokat.

Ezeknek az összetevőknek az együttműködéséből olyan hálózati környezet jön létre, amelyben a számítógépek el tudnak indulni előzetes TCP/IP beállítás nélkül is; kapnak egy helyileg használható (útválasztók számára emészthetetlen) IP címet, regisztrálják gépnevüket a többi (helyi hálózaton működő) számítógépnél, és feltérképezik az elérhető hálózati szolgáltatásokat (például a fájl- és nyomtatókiszolgálókat). Mindez egy olyasféle grafikus felületen keresztül valósul meg, mint amilyen a „Hálózati környezet" (*Network Neighborhood*) – csak rá kell kattintani a kívánt részre egy megfelelő böngészőben.

Az Apple **mDNS** nevű protokollt használ a többes-küldéses DNShez és **DNS-SD**-t (amely a hagyományos DNS rendszer kiterjesztése) a szolgáltatás-felderítéshez.

A Microsoft a többes-küldéses DNS helyett egy alternatív protokollt definiál: Ez a **Link-Local Multicast Name Resolution** (**LLNR** - adott alhálózaton egyedi, többes-küldéses névfeloldás). Szolgáltatás-felderítéshez pedig a **Simple Service Discovery Protocol**-t (**SSDP**; egyszerű szolgáltatás-felderítési protokoll).

Az SSDP (a hagyományos DNS-sel szemben) HTTP-alapú, amely illeszkedik az egyre hangsúlyosabb URL-alapú szolgáltatásokhoz, de megtöri a folyamatosságot a hagyományos DNS infrastruktúrával.

A Microsoft, az Apple és más gyártók is részt vesznek a konfigurációmentes TCP/IP hálózatokról szóló megbeszéléseken, de a nagy cégek némileg eltérő rendszerekhez szeretnék kialakítani megoldásaikat. A legnagyobb különbség a szolgáltatás-felderítési protokollok terén tapasztalható. Vannak más szolgáltatás-felderítő módszerek is, mint például a Szolgáltatás-lokalizáló protokoll (*Service Location Protocol*, SLP), amelyet a HP nyomtatók és sok más eszköz is használ.

Attól még, hogy egy nagyobb operációsrendszer-gyártó egy adott protokoll-opcióhoz tartja magát, az nem jelenti azt, hogy azon az operációs rendszeren csak az használható. Az alkalmazásfejlesztők olyan protokollt használhatnak, amilyet akarnak. Az Apple például kifejlesztette a Bonjour rendszer Windows-os megfelelőjét.

Többféle konfigurációmentes protokoll látott már napvilágot vázlatos RFC-k formájában. Egy ezekhez hasonló rendszer már bekerült az IPv6 terveibe. A következő években minden valószínűség szerint komoly előrelépés várható a konfigurációmentes technológiák fejlődése terén.

Összefoglalás

A DHCP egyszerű megoldást jelent az ügyfélgépek IP címeinek (és más hálózati paramétereinek) beállítására. Különösen hasznos akkor, amikor változások történnek; például, ha internetszolgáltatót váltunk. Ilyenkor kénytelenek vagyunk megváltoztatni TCP/IP beállításainkat. Ha egy olyan intézményben dolgozunk, amelyben (több telephelyen) összesen 5000 gép üzemel, akkor ilyen esetben drága, időigényes és embertelen munka várna a kézi beállításokat elvégző adminisztrátorra. Egy DHCP kiszolgálóval azonban ez a változás egyszerűen megvalósítható, hiszen elegendő a DHCP kiszolgálón bejegyezni a friss adatokat. Amikor a következő alkalommal a DHCP ügyfél meg akarja újítani IP címét, akkor már az új DNS névkiszolgálókra vonatkozó adatokat fogja megkapni.

Óránkon tanultunk a hálózati címfordításról (NAT) és a konfigurációmentes protokollokról is.

Kérdések és válaszok

K *Hogyan kommunikál egymással a DHCP ügyfél és a DHCP kiszolgáló akkor, amikor az ügyfélgép először indul el?*

V Üzenetszórással (*broadcasting*) küldik és fogadják az adatcsomagokat.

K *Mi szükséges ahhoz, hogy az egyik alhálózaton működő DHCP ügyfél egy másik alhálózaton működő DHCP kiszolgálótól IP címet kérhessen?*

V Egy DHCP továbbító ügynök (*relay agent*).

K *Játszhatja-e egy útválasztó a továbbító ügynök szerepét? Bármelyik útválasztó megfelel-e erre a célra?*

V Igen, lehet egy alkalmas útválasztó is a továbbító ügynök. Nem. Erre a célra csak az RFC 1542-kompatibilis útválasztók használhatóak.

K *Hogyan javítja a NAT a hálózati biztonságot?*

V Mivel a NAT cím nem folytonos és nem használható útválasztásra, a külső támadó nem tud kommunikálni a helyi hálózattal. Fontos azonban tudni, hogy ettől még nincs garantálva a belső hálózat biztonsága. A támadók kifejlesztettek olyan módszereket, amelyekkel a NAT mögötti hálózatokhoz is hozzáférést tudnak szerezni.

A fejezetben megismert legfontosabb fogalmak

Ebben a fejezetben a következő kulcsfontosságú fogalmakkal ismerkedtünk meg:

- **Automatic Private IP Addressing (APIPA)** – Automatikus privát IP címzés; Az Ipv4LL Windows-os változata, amely Microsoft rendszereken használatos.
- **BOOTP** – Elsősorban merevlemez nélküli számítógépekhez történő IP cím hozzárendelést lehetővé tevő protokoll.
- **DHCP (Dynamic Host Configuration Protocol ; *dinamikus beállító protokoll*)**. Dinamikus IP cím hozzárendelést lehetővé tevő protokoll.
- **DHCP ügyfél** – Olyan számítógép, amely rendelkezik a TCP/IP hálózatok kezeléséhez szükséges szoftverrel, de nincsenek rajta beállítva a TCP/IP paraméterek.
- **DHCP kiszolgáló** – Olyan számítógép, amely képes a DHCP ügyfelek beállítására. El tudja számukra küldeni az IP címet, az alhálózati maszkot és más TCP/IP hálózati paramétereket is.
- **DNS szolgáltatás-felderítés (*Service Discovery*)** – Lehetőség az ügyfélgépek számára, hogy felderítsék egy konfigurációmentes hálózaton az elérhető szolgáltatásokat.
- **Link Local Addressing** – Adott alhálózaton egyedi címzés. Olyan technológia, amely egy konfigurációmentes hálózaton lehetővé teszi az IP címek beállítását.
- **Multicast DNS** – Többes-küldéses DNS; DNS névfeloldási módszer, amely nem igényel kiszolgálót, sem pedig előre beállított `hosts` állományokat.
- **Zeroconf** – Protokollok gyűjteménye, amely lehetővé teszi bizonyos TCP/IP szolgáltatások használatát egy konfigurációmentes hálózaton.

13. ÓRA

IPv6 – Az új generáció

A fejezet tartalmából:

- Az IPv6 kifejlesztésének okai
- Az IPv6 fejlécformátuma
- Címzés az IPv6-ban

Az Internet folyamatosan változik, így az internetes kommunikációt vezérlő protokolloknak is állandóan változniuk kell. Az Internet protokollt, amely meghatározza a mindenható IP-címzési rendszert, majdnem tíz éve igyekeznek továbbfejleszteni. Ebben az órában azt nézzük meg, hogy mire számíthatunk az IP következő nemzedékében.

Az óra végeztével a következőkre leszünk képesek:

- El tudjuk majd magyarázni, miért van szükség egy új IP-címzési rendszerre.
- Le tudjuk írni az IPv6 fejlécmezőit.
- Alkalmazni tudjuk az IPv6-címek írásának és egyszerűsítésének szabályait.
- Az IPv4-címeket le tudjuk fordítani az IPv6 címterére.

Miért van szükség új IP-változatra?

Az IP címzési rendszere, amelyet a 4. fejezetben ismertettünk, majdnem egy emberöltő óta szolgálja az internetes közösséget, a kifejlesztői pedig méltán lehetnek büszkék arra, hogy milyen messzire jutott a TCP/IP. Az internetes közösségnek azonban van egy nagy gondja: a világ lassan kifogy a szabad címekből. Ez a fenyegető címválság meglepőnek tűnhet, hiszen a jelenlegi IP-formátum 32 bites címmezői több mint hárommilliárd gépet képesek azonosítani. Nem szabad azonban elfelejtenünk, hogy ennek a hárommilliárd címnek a jelentős része valójában nem használható.

Egy cég vagy intézmény jellemzően egy hálózati azonosítót kap, és ők döntik el, hogy a saját hálózatukban milyen gépcímeket osztanak ki. A 4. fejezetből emlékezhetünk rá, hogy az IP-címeket eredetileg úgy tervezték, hogy a címmező első oktettjének értéke által meghatározott címosztályba essenek. A címosztályokat és a hozzájuk tartozó címtartományokat a 13.1. táblázat mutatja a címosztályokon belül lehetséges hálózatok számával és a hálózatokon belül lehetséges állomások számával együtt. Egy B osztályú címhez 65 534 számítógép (állomás) tartozhat, sok B osztályú szervezet azonban nem rendelkezik 65 534 csomóponttal, ezért az elérhető címeknek csak a töredékét osztja ki. A 127 darab A osztályú hálózatnak 16 777 214 cím áll a rendelkezésére, és ezek többsége ugyancsak használaton kívüli. Azt is érdemes megjegyezni, hogy a 16 510 darab A és B osztályú hálózat állítólag már mind foglalt. A fennmaradó C osztályú hálózatok viszont egyenként csak 254 címet tudnak rendelkezésre bocsátani. (Az IP-címek felépítésével a 4. és 5. fejezetben foglalkoztunk részletesebben.)

Szerencsére a hálózati címfordítás (Network Address Translation, NAT) használata csökkentette az internetkész címek iránti igényt, az 5. fejezetben ismertetett CIDR osztály nélküli címzési rendszer pedig otthont talált sok elveszett címnek. Más újabb fejlesztések – például a mobilhálózatok terjedése – ugyanakkor újra nyomás alá helyezte a címteret.

13.1. táblázat *Hálózatok és címek száma az IP-címosztályokban*

Osztály	Első oktett	Hálózatok száma	Lehetséges címek száma hálózatonként
A	0-126	127	16 777 214
B	128-191	16 383	65 534
C	192-223	2 097 151	254

Az Internet elméleti szakemberei már jó ideje tervezik az átállást egy új címzési rendszerre, és mivel a rendszer amúgy is megérett a fazonigazításra, további javításokat, új technológiák beépítését és új szolgáltatások hozzáadását javasolták az IP-hez. Az új rendszer végül az IP 6-os változatában, az IPv6-ban (IP version 6) kristályosodott ki, amelyet néha *IPng*-nek, az *IP következő generációjának* (IP next generation) is nevez-

nek. Az IPv6 jelenlegi leírását az RFC 2460 tartalmazza, amely 1998 decemberében jelent meg. (Az RFC 2460 számára számos előzetes RFC készítette elő a terepet, és újabb RFC-k is vannak, amelyek folytatják az IPv6-tal kapcsolatos kérdések tisztázását.)

Az IPv6-ban az IP-címek formátuma 128 bites címeket követel meg. Ezt a nagyobb címteret részben feltehetőleg az indokolja, hogy akár egymilliárd hálózatot is képes legyen támogatni. Amint azonban ebben a fejezetben megtanuljuk majd, a nagy címméret ahhoz is elég teret ad, hogy bizonyos fokig biztosítsa az IPv4 és IPv6 rendszerű címek összeegyeztethetőségét.

Lássunk néhányat az IPv6 céljai közül:

- **Bővített címzési lehetőségek** – Az IPv6 nem csak több címet biztosít, hanem tovább is fejleszti az IP-címzést. Az IPv6 például több címzési szintet támogat, javít az automatikus beállítási lehetőségeken, valamint jobb támogatást nyújt a *tetszőleges címzéshez* (anycast addressing), ami lehetővé teszi a bejövő adatcsomagoknak, hogy a „legközelebbi" vagy „legjobb" célhoz érkezzenek a lehetséges célpontok csoportjából.
- **Egyszerűbb fejlécformátum** – Az IPv4 fejlécmezői közül néhányat eltávolítottak, míg más mezőket elhagyhatóvá tettek.
- **A bővítmények és kiegészítő lehetőségek jobb támogatása** – Az IPv6 egyes fejlécinformáció nem kötelező bővítményfejlécekben kaptak helyet. Ez a megoldás anélkül növeli a lehetséges információs mezők tartományát, hogy területet pazarolna el a fő fejlécben. A bővítményfejléceket az útválasztók a legtöbb esetben nem dolgozzák fel, ami tovább karcsúsítja az átvitel folyamatát.
- **Folyamcímkézés** – Az IPv6-ban az adatcsomagokat megjelölhetjük, hogy egy adott folyamszinthez rendeljük őket. A *folyamszint* adatcsomagok egy osztályát jelenti, amely specializált kezelőfüggvényeket igényel. Egy valósidejű szolgáltatás folyamszintje például különbözhet egy elektronikus levél folyamszintjétől. A folyamszint-beállítás abban segít, hogy biztosítsuk az átvitel szolgáltatási minőségének minimumát.
- **Továbbfejlesztett hitelesítés és titkosítás** – Az IPv6 bővítményei támogatják a hitelesítést, a bizalmasságot és a különböző adatépség-ellenőrző eljárásokat.

Könyvünk írásának idején az IPv6 már majdnem 10 éve készen állt, mégis csupán egy maroknyi hálózat valósította meg teljes rendszerként. A probléma oka részben az, hogy az új generációra történő átállás egy átmeneti időszakot igényel, amelynek során egyidejűleg kell támogatni az IPv4-et és az IPv6-ot, és amíg az IPv4 működik, a rendszergazdákat nem szorítja semmi arra, hogy abbahagyják a használatát. Jelenleg minden fontosabb operációs rendszer és a legtöbb útválasztó is kínál IPv6-támogatást, a cégek többsége azonban nem vállalja a mindkét rendszer aktív fenntartásával járó többletköltséget (bár lehetséges, hogy egy IPv6-verem alapértelmezés szerint fut).

Még ha egy cég helyi szinten meg is szeretne valósítani egy natív IPv6-os hálózatot, gondot okozhat olyan internetszolgáltatót találni, aki natív IPv6-támogatást nyújt. Az Internet IPv6 szolgáltatást gyakran IPv6-csatornaközvetítőkön (tunnel broker) keresztül lehet elérni. A csatornaközvetítő az IPv6-os csomagokat egy IPv4-es csatornába csomagolja – ez a megoldás valóban IPv6-os kapcsolatot nyújt a végpontokon, de az IPv6 támogatása egy IPv4-es csatornán keresztül csökkenti az IPv6-ba beépített fejlettebb útválasztási és minőségbiztosítási lehetőségek előnyeit.

Létezik egy az IETF-en keresztül hozzáférhető internetes vázlat, amely útitervet határoz meg az IPv6 teljes megvalósításához, 2012. januári céldátummal. A terv szerint az átállási időszak végén az internetszolgáltatóknak *kötelező* lesz IPv6-szolgáltatásokat nyújtaniuk (és *ajánlott* natív IPv6-szolgáltatásokat biztosítaniuk), a cégeknek és intézményeknek pedig *kötelezően* biztosítaniuk kell majd az IPv6-kapcsolatot az Internetre néző kiszolgálóikon, és *ajánlottan* támogatniuk a belső IPv6-kapcsolatokat. A vázlat 2008. augusztusi elévülési idővel jelent meg – mire ez a könyv nyomdába kerül, feltehetőleg elkészül egy frissített változata.

Az IPv6-hálózatokról ugyanakkor egyre gyakrabban esik szó az operációs rendszerek leírásában és a hozzájuk készült tanmenetekben. Ha a jelenlegi tervezetben felvázolt átállás sikeresnek bizonyul, könyvünk következő kiadása már bizonyára az IPv6-ról fog szólni, és az IPv6 nem a 13., hanem a 4. órában kerül elő. Addig is, ezen az órán áttekintjük az IPv6-tal kapcsolatos legfontosabb fogalmakat.

Az IPv6 fejlécformátuma

Az IPv6 fejlécformátumát a 13.1. ábrán láthatjuk. Figyeljük meg, hogy az alapszintű IPv6-os fejléc valójában egyszerűbb a neki megfelelő IPv4-fejlécnél. A fejléc egyszerűségének részben az az oka, hogy a részletesebb információkat különleges bővítményfejlécekbe száműzték, amelyek a fő fejlécet követik.

13.1. ábra
Az IPv6 fejlécszerkezete

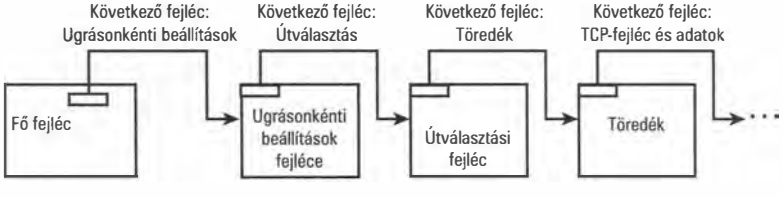
Az IPv6-fejléc mezői a következők:

- **Változat (Version, 4 bit)** – Az IP változatszámát azonosítja (ami ebben az esetben a 6).
- **Forgalomosztály (Traffic Class, 8 bit)** – Az adatcsomagba zárt adatok típusát azonosítja.
- **Folyamcímke (Flow Label, 20 bit)** – A folyamszintet jelöli ki (lásd az előző szakaszt).
- **Értékes hossz (Payload Length, 16 bit)** – Az adatok (az adatcsomagnak a fejléc után következő része) hosszát határozza meg.
- **Következő fejléc (Next Header, 8 bit)** – Az adott fejlécet közvetlenül követő fejléc típusát határozza meg (lásd a bővítményfejlécek tárgyalását ennek a szakasznak a későbbi részében).
- **Ugrásszám (Hop Limit, 8 bit)** – Azt jelzi, hogy még hány ugrás engedélyezett az adott adatcsomag számára. Ez az érték minden ugrással csökken, és ha eléri a nullát, a rendszer elveti az adatcsomagot.
- **Forráscím (Source Address, 128 bit)** – Az adatcsomagot küldő számítógép IP-címét azonosítja.
- **Célcím (Destination Address, 128 bit)** – Az adatcsomagot fogadó számítógép IP-címét azonosítja.

Ahogy ebben az órában már említettük, az IPv6 külön bővítményfejlécekben, amelyek a fő fejléc és az adatok között foglalnak helyet, kiegészítő információcsomagok mellékelését is lehetővé teszi. Ezek a bővítményfejlécek konkrét helyzetekhez nyújtanak információt, miközben lehetővé teszik, hogy a fő fejléc kicsi és könnyen kezelhető maradjon.

Az IPv6-szabvány az alábbi bővítményfejléceket határozza meg:

- Ugrásonkénti beállítások (Hop-by-Hop Options)
- Célbeállítások (Destination Options)
- Útválasztás (Routing)
- Töredék (Fragment)
- Hitelesítés (Authentication)
- Titkosított biztonsági tartalom (Encrypted Security Payload)



13.2. ábra

A „Következő fejléc" mező

Mindegyik fejléctípushoz egy 8 bites azonosító társul. A Következő fejléc mező a fő fejlécben vagy egy bővítményfejlécben a lánc következő fejlécének azonosítóját határozza meg (lásd a 13.2. ábrát).

A fentiekben leírt bővítményfejlécek közül az átviteli útvonalon található köztes csomópontok csak az Ugrásonkénti beállítások és az Útválasztás fejléceket dolgozzák fel. Az útválasztóknak a többi bővítményfejlécet nem kell feldolgozniuk; elég csak átadniuk azokat.

Az alábbiakban mindegyik bővítményfejléc-típust részletesebben is bemutatjuk.

Ugrásonkénti beállítások fejléc

Az Ugrásonkénti beállítások fejléc célja, hogy kiegészítő információkat közöljön az átviteli útvonalon található útválasztókkal. Az Ugrásonkénti beállítások fejléc a következő szakaszban tárgyalt Célbeállítások fejléchez hasonlóan nagyrészt azért került bele a szabványba, hogy a jövőbeli lehetőségek kifejlesztéséhez formátumot és eljárást biztosítson az iparágnak.

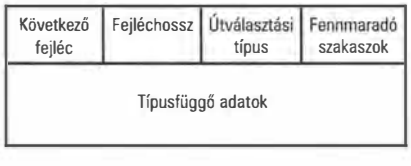
A szabvány szerint a fejléc része a beállítás típusának jelölése, valamint néhány kitöltési beállítás az adatok igazításához. A szabvány által konkrétan meghatározott egyik beállítás a *nagy méretű tartalom* (jumbo payload), amelyet a 65 535 bájtnál hosszabb értékes adatok átvitelére használhatunk.

Célbeállítások fejléc

A Célbeállítások fejléc célja, hogy kiegészítő információkat közöljön a célcsomóponttal. Az Ugrásonkénti beállítások fejléchez hasonlóan a Célbeállítások fejlécet is elsősorban jövőbeli lehetőségek keretrendszerének szánták.

Útválasztás fejléc

Az Útválasztás fejlécben egy vagy több útválasztót határozhatunk meg, amelyeken az adatcsomagnak át kell haladnia a célja felé. Az Útválasztás fejléc formátumát a 13.3. ábra mutatja.



13.3. ábra
Az Útválasztás fejléc

Az Útválasztás fejléc adatmezői a következők:

- **Következő fejléc (Next Header)** – Az adott fejlécet követő fejléc típusát azonosítja.
- **Fejléchossz (Header Length, 8 bit)** – A fejléc hosszát adja meg bájtban (a Következő fejléc mező nélkül).
- **Útválasztási típus (Routing Type, 8 bit)** – Az útválasztási fejléc típusát azonosítja. A különböző útválasztási fejléctípusokat más-más helyzetekhez tervezték.
- **Fennmaradó szakaszok (Segments Left)** – A kifejezetten meghatározott útválasztási szakaszok számát jelzi a cél előtt.
- **Típusfüggő adatok (Type-Specific Data)** – Az Útválasztási típus mezőben megadott útválasztási típus adatmezőit azonosítja.

Töredék fejléc

Egy adott üzenettovábbítási útvonalon minden útválasztó rendelkezik MTU beállítással (maximum transmission unit, legnagyobb átviteli egység). Az MTU beállítás az útválasztó által továbbítható legnagyobb adategységet adja meg. Az IPv6-ban a forráscsomópont felderítheti az *útvonal-MTU*-t, ami az átviteli útvonalon található eszközök legkisebb MTU beállítását jelenti. Az útvonal-MTU tehát a legnagyobb adategységet határozza meg, ami az adott útvonalon továbbítható. Amennyiben az adatcsomag mérete meghaladja az útvonal-MTU értékét, a csomagot kisebb darabokra kell tördelni, hogy átvihető legyen a hálózaton. A széttört adatcsomagok újbóli összeállításához szükséges információkat a Töredék fejléc tartalmazza.

Hitelesítés fejléc

A Hitelesítés fejléc biztonsági és hitelesítési információkat tartalmaz. A Hitelesítés mező segítségével megállapítható, hogy az adatcsomag módosult-e az átvitel közben.

Titkosított biztonsági tartalom fejléc

A Titkosított biztonsági tartalom fejléc (Encrypted Security Payload, ESP) a titkosítást és a bizalmasságot biztosítja. Az IPv6 ESP fejlécének segítségével az átvitt adatok részben vagy egészben tiitkosíthatók. Csatorna módú ESP használata esetén a teljes IP-adatcsomag titkosított lesz, és egy külső, titkosítatlan adatcsomagba kerül, míg átviteli módú ESP használatakor a hitelesítési adatok és az ESP-fejlécinformációk titkosítására kerül sor.

Címzés az IPv6-ban

Az IPv6-címeket az IPv4-címekhez hasonlóan egy központi internethatóság osztja ki az internetszolgáltatók és más sávszélesség-szolgáltatók rendszerén keresztül. Ahogy a 13.2. táblázatban láthatjuk, bizonyos címtartományokat konkrét célokra tartanak fenn, például csoportos vagy többcímes (multicast) vagy kapcsolathoz kötött (link-local) címzésre (ami az IPv4-nek a 12. órában megismert zéró konfigurációs rendszeréhez hasonlít). Ahogy a következőkben megtanuljuk, az IPv4-címek leképezéséhez az IPv6 címterére egy másik különleges címtartományt tartanak fenn.

13.2. táblázat *Az IPv6 címtartományai az RFC 4291 szerint*

Címtípus	Bináris előtag	IPv6-jelölés	Leírás
Nem meghatározott	0..00 (csupa nulla)	::/128	Soha nem szabad kiosztani. A cím hiányát mutatja.
Visszacsatoló	0..01 (127 nulla)	::1/128	Vizsgálati cím, amelyet arra használnak, hogy egy állomás önmagának küldhessen csomagot.
Leképezett IPv4	0..0:FFFF (80 nulla)	::FFFF/96	Egy meglevő IPv4-es cím IPv6-os megfelelője.
Csoportos	11111111	FF00::/8	Állomások egy csoportját azonosítja.
Kapcsolathoz kötött	1111111010	FE80::/10	Automatikus címbeállításhoz használják.
Globális egycímes (minden más)			

Akárhogy is nézzük, a 128 bites IPv6-címek jelentős terhet rónak a memóriára. A 4. óráról emlékezhetünk rá, hogy a 32 bites IPv4-címeket általában pontozott decimális jelöléssel szokták feltüntetni, amelyben minden bájtot egy legfeljebb három számjegyből álló decimális szám ábrázol. Ezt a 12 decimális számjegyből álló karakterláncot sokkal könnyebb áttekinteni, mint a tényleges bináris cím 32 bináris számjegyét – sőt kis erőfeszítéssel a pontozott decimális címeket akár meg is lehet jegyezni. A 32 bites címek emberivé tételének ez a módja azonban teljesen hasztalan, ha 128 bites címekre szeretnénk emlékezni. Nem csoda, hogy több módszert is kidolgoztak az ijesztő IPv6-címek egyszerűsítésére. Az IPv6-címeket általában nyolc, kettősponttal elválasztott oszlopban ábrázolják, amelyek mindegyike négy hexadecimális (tizenhatos számrendszerű) számjegyet tartalmaz, a bevezető nullák elhagyásával:

```
2001:DB8:0:0:8:800:200C:417A
```

Egy gyorsírásos trükk az egymást követő, nullákból álló blokkok kiküszöbölésére, hogy két kettőspontra cserélik őket. A fenti cím ebben az esetben így írható fel:

```
2001:DB8::8:800:200C:417A
```

Minden címben csak egy kettős kettőspont lehet. Az IPv6-címek hozzárendelési szabályai gyakran eredményeznek hosszú, nulla bitekből álló karakterláncokat, ami a kettős kettőspontokat különösen hasznossá teszi. Nézzük például ezt a címet:

```
FF01:0:0:0:0:0:0:101
```

Ezt egyszerűen így is felírhatnánk:

```
FF01::101
```

Az IPv4-címekhez hasonlóan az IPv6-címek is egy előtaggal kezdődnek, ami a hálózatot jelképezi. A CIDR rendszer (lásd az 5. órát) megfelelője egy címtömb leírását is lehetővé teszi, a tömb első címének meghatározásával, valamint egy decimális számmal, ami a hálózati bitek számát jelöli. Az RFC 4291 (*IPv6 Addressing Architecture*, Az IPv6 címzési rendszere) szerint a 20010DB80000CD3 60 bites hálózati előtaggal rendelkező címek tömbjét így írhatjuk fel:

```
2001:0DB8:0000:CD30:0000:0000:0000:0000/60
```

Vagy:

```
2001:0DB8:0:CD30::/60
```

Az IPv6-os hálózati beállítóprogramok lehetővé teszik majd a felhasználónak, hogy meghatározzon egy alapértelmezett hálózati előtagot, hogy az ügyfélen végzett kézi beállításnál csak a cím állomás (host) részére kelljen hivatkozni. Az IPv6 emellett kifinomult automatikus beállítási lehetőségeket is nyújt, ami csökkenti a hosszú címek begépelésének szükségességét.

Még túl korai lenne jóslatokba bocsátkozni arról, hogy a hálózati rendszergazdák hogyan alkalmazkodnak majd az irdatlan IPv6-címekhez, de az biztosra vehető, hogy a névfeloldás fontos szerepet fog játszani az IPv6-hálózatokon (lásd a 11. fejezetet).

IPv6 az IPv4 mellett

Természetesen az IPv6 egyetlen esélye arra, hogy teret nyerjen, ha fokozatosan vezetik be. Az Internet teljes körű átszervezése nem fog bekövetkezni, ezért a fejlesztők úgy tervezték meg az IPv6-ot, hogy egy hosszú átmeneti időszakban képes legyen együtt élni az IPv4-gyel. Az elképzelés az, hogy az IPv4 protokollverme mellett egy IPv6-protokollverem fog működni többprotokollos rendszerben, ugyanúgy, ahogy az IPv4 valaha együtt élt az IPX/SPX-szel, a NetBEUI-val és más protokollvermekkel.

Az IPv6 címzési rendszere lehetőséget nyújt a meglevő IPv4-címek leképezésére az IPv6 címterére. Az eredeti terv az volt, hogy minden érvényes IPv4 címet egy 128 bites IPv6-címre fordítanak, egyszerűen 96 nulla bitet téve az eredeti cím elé. Ez az alak, amelyet

IPv4-megfelelő IPv6-címként ismernek, az RFC 4291 megjelenésével elavulttá vált, és átadta a helyét egy másik módszernek, az *IPv6-ra leképezett IPv4-címek* használatának, amelyek 80 nulla bitből, majd 16 egy bitből (hexadecimális formában FFFF), majd az eredeti 32 bites IPv4-címből állnak.

Vegyük például az alábbi IPv4-címet:

```
169.219.13.133
```

Ez a következő IPv6-címre képeződik le:

```
0000:0000:0000:0000:0000:FFFF:A9DB:0D85
```

Vagy egyszerűen:

```
::FFFF:A9DB:0D85
```

Mivel az előtag világosan a leképezett IPv4-címek tartományában helyezi el a fenti címet, az IPv4-részt néha meghagyják az ismerős pontozott decimális alakban:

```
::FFFF:169.219.13.133
```

Az IPv6 és a szolgáltatás minősége (QoS)

Az IPv6 megoldást nyújt egy másik kihívásra is, amellyel a korosodó IPv4 rendszer nemrégiben került szembe: az egységes szolgáltatásminőségi szintek (Quality of Service, QoS) szükségességére. Régen, amikor az Internetet elsősorban elektronikus levelezésre és FTP-stílusú letöltésre használták, senki sem törődött különösebben az adatátvitel elsőbbségének biztosításával. Ha egy e-mail nem érkezett meg 2 másodpercen belül, akkor megérkezett 2 perc – vagy akár egy óra – alatt.

Senki sem fáradt azzal, hogy meghatározza vagy korlátozza az üzenet megérkezésére rendelkezésre álló időtartamot. Ezzel szemben ma az Internet az átvitel számos típusát támogatja, amelyek némelyike szigorú követelményeket támaszt a kézbesítéssel szemben. Az internetes videó-, tévé- és más valósidejű alkalmazások nem képesek rendeltetésszerűen működni, ha a csomagok csak hosszas késleltetéssel érkeznek meg az útválasztók átmeneti tárainak útvesztőjében bolyongva. Egy internettelefonos kapcsolatban még egy kis késleltetés is a résztvevők hangjának torzulását eredményezheti.

A jövő Internete lehetővé teszi majd a kézbesítésre váró IP-adatcsomagok rangsorolását. Egy interaktív videóalkalmazástól származó adatcsomag a várakozási sor elejére kerülhet, miközben egy útválasztó átmeneti tárára vár, míg egy e-mail adatcsomagja kis ideig várakozhat.

Az IPv6-ot úgy tervezték, hogy különböző szolgáltatási szinteken keresztül támogassa a rangsorolást. Az adatcsomagba zárt adatok típusának és elsőbbségének meghatározását az IPv6 fejléc Forgalomosztály és Folyamcímke mezői teszik lehetővé (lapozzunk vissza a 13.1. ábrához).

Egyes gyártók és mérnökök az IPv4 Szolgáltatástípus (Type of Service) mezőjével kísérleteznek a megkülönböztetett szolgáltatási információk tárolására. Az IPv6 Forgalomosztály mezője a tervezők szándéka szerint továbbra is támogatja a kísérletezést a megkülönböztetett szolgáltatásokkal.

Összefoglalás

Az IPv6, az IP protokoll következő generációja, lassan utat tör magának a való világba. Az IPv6 címzési rendszere teljesen különbözik a 4. órán megismert rendszertől. 128 bites címtere szinte korlátlan számú címnek adhat otthont; ezen kívül az IPv6 egyszerűbb fejléceket biztosít, nagyobb értékes tartalmat tesz lehetővé, és számos továbbfejlesztést tartalmaz a biztonság és a szolgáltatás minősége terén.

Kérdezz–felelek

K *Miért van olyan sok IP-cím használaton kívül?*

V Egy cég vagy szervezet, amely internetes címtérrel rendelkezik, gyakran nem használja az adott címtér összes lehetséges gépcímét.

K *Mi az előnye annak, ha a fejlécinformációkat egy bővítményfejlécbe helyezzük a fő fejléc helyett?*

V A bővítményfejlécet csak akkor kell mellékelni, ha a benne található információkra szükség van. Ezenkívül az útválasztók számos bővítményfejlécet nem dolgoznak fel, ezért azok nem lassítják le az útválaszó forgalmát.

K Hogyan segíti majd az IPv6 az olyan valósidejű alkalmazások működését, mint a videókonferenciák?

V Az IPv6 fejléc Forgalomosztály és Folyamcímke mezői lehetővé teszik az adatok típusának és elsőbbségének meghatározását.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Értékes hossz** – Egy IPv6-fejléc adatrészének hossza (a fejlécet leszámítva).
- **Folyamszint** – Egy IPv6-os adatcsomag megjelölése, amely különleges kezelést vagy különleges szintű (például valós idejű) átvitelt ír elő.
- **IPv6** – Az IP-címzés új szabványa, amely 128 bites IP-címeket használ. Az IPv6 tervezőinek célja az, hogy az IPv6 a következő néhány évben fokozatosan átvegye az uralmat.
- **Letölthető munkaterület** – Hordozható, egyedi munkakörnyezet, amelyet a felhasználó attól függetlenül érhet el, hogy honnan jelentkezett be.
- **Nagy méretű tartalom (Jumbo payload)** – Olyan értékes tartalom egy adatcsomagban, amelynek a hossza meghaladja a 65 535 bájt szokásos korlátját. Az IPv6 a nagy méretű tartalommal rendelkező adatcsomagoknak is lehetővé teszi, hogy áthaladjanak a hálózaton.
- **Pontozott decimális jelölés** – A 32 bites IP-címek decimális megfelelőjének szokványos formátuma (például: `111.121.131.144`).
- **Ugrásszám** – Az útválasztók közötti ugrások fennmaradó száma – legfeljebb ennyit ugorhat az adatcsomag, mielőtt a rendszer elvetné. Az ugrásszámot az IPv6-ban a fő fejléc határozza meg, és az értéke minden alkalommal csökken, amikor az adatcsomag egy útválasztóhoz ér.

IV. RÉSZ

TCP/IP-eszközök

14. ÓRA

TCP/IP-eszközök

A fejezet tartalmából:

- Protokollproblémák
- Vonalproblémák
- Névfeloldási problémák
- Problémák a hálózat teljesítményével
- Az FTP és az SFTP
- A TFTP
- Az RCP és az SCP
- Az NFS és az SMB fájlszolgáltatási protokollok

A TCP/IP környezet számos szabványos segédprogramot tartalmaz a hálózati kapcsolatok beállítására, kezelésére és hibaelhárítására, más eszközök pedig olyan feladatok elvégzését könnyítik meg, mint a kommunikáció vagy a fájlátvitel. Ezeknek a TCP/IP-segédprogramoknak a története még a modern grafikus felhasználói felületek születése előttre nyúlik vissza, így sokukat parancssoros használatra tervezték. A parancssoros felület régimódinak tűnhet, de sok tapasztalt hálózati rendszergazda még mindig gyorsabbnak, egyszerűbbnek és hatékonyabbnak tartja a parancssorból végzett munkát, mint az egérrel való kattintgatást és az ugrálást az ablakok között.

Ezt az órát néhány olyan segédprogram bemutatásával kezdjük, amelyeket a TCP/IP beállításához és hibaelhárításához használhatunk. Ezeket az eszközöket nélkülözhetetlennek fogjuk találni, amikor kapcsolati problémákat kell azonosítanunk, tesztelnünk kell a hálózati csomópontok közötti kommunikációt, vagy ellenőriznünk kell a hálózatunk számítógépein a TCP/IP-beállításokat.

Ezt követően megismerkedünk néhány fontos fájlhozzáférési TCP/IP-eszközzel is, köztük az FTP-vel (File Transfer Protocol, fájlátviteli protokoll), az SFTP-vel (Secure File Transfer Protocol, biztonságos fájlátviteli protokoll), a TFTP-vel (Trivial File Transfer Protocol, egyszerű fájlátviteli protokoll) és az `rcp`-vel (Remote Copy, távmásolás).

Az óra végeztével a következőkre leszünk képesek:

- Meg tudjuk határozni és le tudjuk írni a fontosabb kapcsolati TCP/IP-segédprogramokat.
- Használni tudjuk a kapcsolati segédprogramokat hibák elhárításához.
- El tudjuk magyarázni az FTP és az SFTP célját, illetve használatát.
- Kezdeményezni tudunk FTP-munkameneteket, az FTP-parancsok segítségével be tudunk járni távoli könyvtárszerkezeteket, fájlok tudunk cserélni távoli rendszerekkel, és létre tudunk hozni, illetve el tudunk távolítani könyvtárakat.
- El tudjuk magyarázni a TFTP célját, illetve használatát.
- Fel tudunk építeni egy fájlátviteli parancsot a TFTP segítségével.
- El tudjuk magyarázni az `rpc` és az `scp` célját, illetve használatát.

Kapcsolati problémák

Ahogy a korábbi órák során megtanultuk, a protokoll egy kommunikációs szabvány. Ezt a szabványt aztán egy szoftvergyártó valósítja meg egy szoftvermodulban, amely a szabványban leírt műveleteket hajtja végre. Az ember telepíti és beállítja a protokollszoftvert, vagy közvetlenül, vagy egy olyan operációs rendszert telepítve, amely támogatja azt. Ahogy kitalálhatjuk, előfordulhat, hogy bár a szoftver működésre készen áll, a hálózat mégsem működik. Néha bizonyos szolgáltatások működnek, míg mások nem, máskor pedig a számítógép nem minden távoli PC-vel tud kapcsolatot létesíteni. Időnként az is megesik, hogy a számítógép látszólag semmilyen hálózati hozzáféréssel nem rendelkezik, mintha nem is lenne csatlakoztatva.

A hálózati problémák jellemzően néhány okra vezethetők vissza. A TCP/IP-közösség több segédprogramot is kifejlesztett ezeknek a problémáknak a felderítésére. Ezen az órán a leggyakoribb hálózati problémákat és a megoldásukra szolgáló eszközöket mutatjuk be.

A leggyakoribb hálózati kapcsolati problémák általában az alábbi négy típus variációi:

- **Hibás protokollműködés vagy -beállítás** – A protokollszoftver nem működik, vagy (valamilyen okból kifolyólag) helytelenül van beállítva a megfelelő működéshez az adott hálózaton.
- **Vonalproblémák** – Egy kábel nincs bedugva, vagy nem működik, esetleg egy elosztó (hub), útválasztó (router) vagy kapcsoló (switch) működésével vannak gondok.
- **Hibás névfeloldás** – A DNS- vagy NetBIOS-nevek nem oldhatók fel. Az erőforrások az IP-címükön elérhetők, de állomásnévvel vagy DNS-névvel (például `www.sun.com`) nem.
- **Túlzott adatforgalom** – A hálózat működőképesnek tűnik, de nagyon lassú.

A következőkben olyan eszközöket és eljárásokat mutatunk be, amelyek ezeknek a szokványos kapcsolati problémáknak az elhárításában segítenek.

Hibás protokollműködés vagy -beállítás

Mint minden szoftvernél, a TCP/IP-protokollszoftver esetében is előfordulhat, hogy nem megfelelően települ; de még ha megfelelően telepítettük is, megeshet, hogy egy sérült fájl vagy a rendszerbeállítások valamilyen változása miatt működésképtelenné válik. Lehetséges például, hogy a szoftver működik, de a számítógép nem tud más számítógépekhez kapcsolódni, mert az IP-címét vagy az alhálózati maszkját helytelenül állították be.

A TCP/IP-protokollcsomag számos hasznos segédprogramot tartalmaz, amelyek segítenek felderíteni, hogy a TCP/IP működőképes-e, illetve hogy helyesen lett-e beállítva:

- **`ping`** – Ez a segédprogram egy rendkívül hasznos diagnosztikai eszköz, amely egyszerű visszhangkéréssel ellenőrzi a hálózati kapcsolatot, és jelentést ad arról, hogy a másik számítógép válaszolt-e.
- **Beállításinformációs segédprogramok** – Minden operációsrendszer-gyártó biztosít valamilyen segédprogramot, amellyel megjeleníthetjük a TCP/IP-beállításokat, és ellenőrizhetjük, hogy az IP-cím, az alhálózati maszk, a DNS-kiszolgáló és más paraméterek beállítása megfelelő-e.
- **`arp`** – Ez a segédprogram az IP-címeket fizikai (MAC-) címekhez társító ARP-gyorsítótár (lásd a 4. fejezetben) tartalmának megtekintését és beállítását teszi lehetővé.

Ezek a segédprogramok minden operációs rendszeren a TCP/IP-megvalósítás szabványos részét képezik. A következőkben részletesen is megvizsgáljuk őket.

ping

Ha azt vesszük észre, hogy a számítógépünk nem tud befejezni egy hálózati műveletet, az első kérdés, amelyet fel kell tennünk, hogy képes-e bármilyen más hálózati műveletet végrehajtani. Más szavakkal, a számítógépünk jelenleg a hálózat tagjaként működik? A `ping` segédprogram a lehető legegyszerűbb hálózati kapcsolati tesztet hajtja végre: olyan üzenetet küld egy másik számítógépnek, ami csak annyit kérdez, hogy „Ott vagy?", és várja, hogy a másik számítógép válaszoljon.

A *ping* kifejezés a tengeralattjárók és más hajók által a különféle objektumok észlelésére használt szonártechnológiából ered, de maga a `ping` a Packet Internet Groper rövidítése.

A ping alapvető utasításformája a következő:

```
ping <IP_cím>
```

Az *IP_cím* annak a számítógépnek a címe, amelyhez kapcsolódni szeretnénk. Más segédprogramokhoz hasonlóan a `ping` is kínál néhány további parancssori kapcsolót, de ezek a megvalósítástól és az operációs rendszertől függően eltérőek lehetnek.

A `ping` segédprogram a címzett számítógépnek az ICMP Echo Request (visszhangkérés) parancs segítségével küld üzenetet (az ICMP-ről a 4. fejezetben beszéltünk bővebben). Amennyiben a címzett számítógép jelen van, és működik, az ICMP Echo Reply (visszhangválasz) üzenettel válaszol. Amikor a küldő számítógép megkapja a választ, kiír egy üzenetet, amely jelzi, hogy a visszhangkérés sikeres volt.

A `ping` parancs sikeres végrehajtása megerősíti, hogy mind a visszhangkérő („pingelő"), mind a válaszoló („megpingelt") számítógép a hálózaton van, és képes kommunikálni. Ne feledjük azonban, hogy a `ping` egy minimális alkalmazás, amely csak a TCP/IP-verem alsó két rétegének működőképességét követeli meg. A `ping` akkor is működik, ha a TCP-vel, az UDP-vel vagy a felső két rétegben található alkalmazásokkal van gond. Ha a `ping` megfelelően működik, általában az olyan elemek hibáját zárhatjuk ki, mint a hálózati hozzáférési réteg, a hálózati kártya, a kábelek vagy az útválasztók.

A `ping` több olyan lehetőséget is kínál, amelyek különösen hasznossá teszik, ha hálózati problémákat kell elhárítanunk. Megtehetjük például a következőket:

- Egy különleges IP-címet, az úgynevezett visszacsatolási címet (loopback) megadva visszhangkérést intézhetünk a helyi IP-szoftverhez. Ez a cím a `127.0.0.1`. Ha a `ping 127.0.0.1` parancs sikeres, a TCP/IP-protokollszoftverünk megfelelően működik.
- Visszhangkérést intézhetünk a saját IP-címünkhöz (vagyis „megpingelhetjük magunkat"). Ha a hálózati kártyánkhoz rendelt IP-címről választ kapunk, akkor tudhatjuk, hogy a csatoló beállítása megfelelő, és kommunikál a TCP/IP-szoftverrel.

- Állomásnév szerint is végrehajthatunk visszhangkérést. A legtöbb rendszer megengedi, hogy az IP-cím helyett egy állomásnevet adjunk meg a `ping` parancsban. Ha egy számítógépet IP-cím szerint el tudunk érni a `ping`-gel, de az állomásnevével nem, akkor tudhatjuk, hogy a probléma a névfeloldással van.

A hibák elhárítása során a hálózati rendszergazda általában a következő `ping`-parancsokat hajtja végre:

1. Visszhangkérést intéz a visszacsatolási címhez (`127.0.0.1`), hogy ellenőrizze, hogy a TCP/IP megfelelően működik-e a helyi számítógépen.
2. Visszhangkérést intéz a helyi IP-címhez, hogy ellenőrizze, hogy a hálózati kártya megfelelően működik-e, és hogy be van-e állítva a helyi IP-cím.
3. Visszhangkérést intéz az alapértelmezett átjáróhoz, hogy ellenőrizze, hogy a számítógép képes-e kommunikálni a helyi alhálózattal, illetve hogy az alapértelmezett átjáró üzemel-e a hálózaton.
4. Visszhangkérést intéz egy címhez az alapértelmezett átjárón túlra, hogy ellenőrizze, hogy az átjáró sikeresen továbbítja-e a csomagokat a helyi hálózati szakaszon kívülre.
5. Visszhangkérést intéz állomásnév szerint a helyi állomáshoz, illetve távoli állomásokhoz, hogy ellenőrizze, hogy a névfeloldás megfelelően működik-e.

A fenti lépések jó kezdetet jelentenek a hálózati problémák felderítéséhez. Lehet, hogy a hiba forrását nem sikerül felderítenünk, de legalább tájékozódási pontot kapunk, hogy hol keressük.

A ping kimenetének közelebbi vizsgálata

A `ping` parancs kimenete a megvalósítástól függően változhat. Egyes rendszereken, például a Solarison, a kimenet egyetlen sor, amely annyit mond, hogy az *<ip_cím>* `is alive` („az *<ip_cím>* él"). A Linux egyes változatai (alapértelmezés szerint) folyamatosan küldik az ICMP-csomagokat, és írják ki a válaszinformációkat, amíg le nem nyomjuk a CTRL+C billentyűket. A Windows rendszerek jellemzően négy ICMP-visszhangkérést bocsátanak ki, és az ezekhez tartozó válaszokat jelenítik meg. Nem szokatlan, ha a négy visszhangkérésre csak három vagy annál is kevesebb válasz érkezik. Az időnként elvesztett adatcsomagokat nem kell hibának tekintenünk, mert az ICMP protokoll nem garantálja a kézbesítést, ugyanakkor a hiányzó válaszok túlzsúfolt hálózatra utalhatnak. A néha elvesztett csomagok ellenére a `ping`-re kapott válasz a leggyakrabban az, hogy minden visszhangkérés sikeres volt (ami azt jelzi, hogy a kapcsolat működik), vagy valamennyi kudarcot vallott (mely esetben a kapcsolat nem működik). A `ping` segédprogram egyes változatai az Echo Request üzenet kibocsátásától az Echo Reply üzenet beérkezéséig eltelt időt is megjelenítik, ezredmásodpercben. A rövid válaszidők arra utalnak, hogy az adatcsomagoknak nem kell túl sok útválasztón vagy lassú hálózatokon áthaladniuk. Ha a visszhangkérésre nullához közeli TTL-értéket kapunk, az azt jelezheti, hogy a kapcsolat közelíti a TTL-határértéket, és egyes csomagok elveszhetnek vagy újra kell küldeni azokat.

Beállításinformációs segédprogramok

Minden modern operációs rendszer kínál olyan segédprogramot, amely lehetővé teszi az érvényben levő TCP/IP-beállítások megtekintését. Ezek a segédprogramok olyan adatokat írnak ki, mint az adott (helyi) számítógép IP-címe és alhálózati maszkja vagy az alapértelmezett átjáró. A segítségükkel meggyőződhetünk róla, hogy a számítógép IP-címinformációi megfelelnek-e a várakozásainknak. A DHCP népszerűvé válásával a beállítófájlokból vagy a beállítási párbeszédablakokból nem mindig tudjuk meghatározni az IP-címinformációkat, a beállításinformációs segédprogramok azonban elárulják, hogy a számítógép éppen milyen címet használ. Ha a számítógépünk a DHCP használatára van beállítva, még azt is felfedezhetjük, hogy a gépnek egyáltalán nincs IP-címe, ami a DHCP-kiszolgálóval való kapcsolati hibára utal.

Természetesen ezek a segédprogramok azt nem árulják el, hogy milyen IP-címet és alhálózati maszkot kellene használnunk, csupán azt, hogy a számítógép éppen milyen címeket alkalmaz. A mi feladatunk, hogy ezeknek az információknak a birtokában megállapítsuk, hogy a címparaméterek megfelelnek-e a hálózatunk IP-címzési sémájának (lásd az 5. és 6. fejezetet).

Unix és Linux rendszereken az `ifconfig` parancsot használhatjuk a címinformációk megjelenítésére. Ahogy emlékezhetünk rá az előző órákról, az IP-cím valójában egy hálózati felülethez (például egy hálózati csatolókártyához) társul, nem pedig magához a számítógéphez. Amennyiben a számítógép két hálózati felülettel rendelkezik, két IP-címe lesz. Az `ifconfig` parancs minden hálózati felülethez tartozó címinformációt megjelenít.

Ha meg szeretnénk jeleníteni az IP-címinformációkat az `ifconfig` paranccsal, a következőt kell beírnunk:

```
ifconfig <felület_neve>
```

Itt a *<felület_neve>* annak a hálózati csatolófelületnek a neve, amelyről címinformációkat szeretnénk kapni. (Unix és Linux rendszereken minden hálózati felület egy nevet kap a felületet meghatározó beállítófájltól, és ezzel a névvel hivatkozhatunk rá.) Az alábbi parancs például az `eth0` nevű felülethez tartozó aktuális IP-címet és hálózati maszkot (valamint a használt Unix/Linux-változattól függően egyéb paramétereket) jeleníti meg:

```
ifconfig eth0
```

Az `ifconfig` azt is lehetővé teszi, hogy közvetlenül adjuk meg egy hálózati felület IP-címinformációit, az IP-címet és a hálózati maszkot közvetlenül a parancssorba beírva:

```
ifconfig eth0 <IP_cím> netmask <hálózati_maszk>
```

Itt az *<IP_cím>* a felület címe, a *<hálózati_maszk>* pedig a felület hálózati maszkja.

Az `ifconfig` parancs `up` és `down` kapcsolói a hálózati felület engedélyezésére, illetve letiltására szolgálnak. Például:

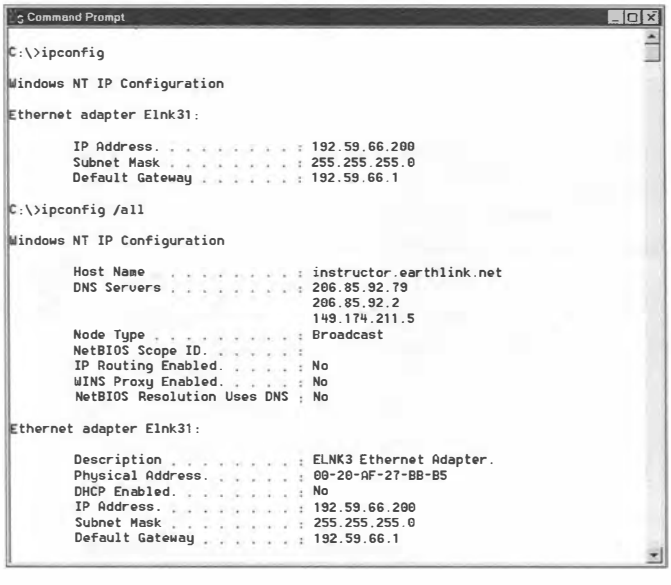
```
ifconfig eth0 up
ifconfig eth0 down
```

Az `ifconfig` parancsnak más kapcsolói is vannak; ezek változatonként különböznek. Ha többet szeretnénk tudni az `ifconfig` parancsról, olvassuk el a parancs Unix/Linux-súgóoldalát:

```
man ifconfig
```

Windows rendszereken a helyi TCP/IP-beállítások megjelenítéséhez az `ipconfig` parancsot használhatjuk. Ha listát szeretnénk kapni az `ipconfig` kapcsolóiról, írjuk be az `ipconfig /?` parancsot. A fontosabb kapcsolók között az alábbiakat találjuk:

- **Alapértelmezett (kapcsolók nélkül)** – Ha az `ipconfig` parancsot kapcsolók nélkül használjuk, az egyes beállított felületekhez tartozó IP-címet, alhálózati maszkot és alapértelmezett átjárót jeleníti meg (lásd a 14.1. ábra felső részét).
- **`all`** – Az `ipconfig /all` parancs olyan további információkat jelenít meg, mint a használatra beállított DNS- és WINS-kiszolgáló(k) IP-címe, valamint a helyi hálózati csatolókártyába beégetett fizikai cím (MAC-cím). Ha a címeket egy DHCP-kiszolgálótól kapjuk, az `ipconfig` a DHCP-kiszolgáló IP-címét mutatja meg, valamint a címbérlet lejárati idejét.



14.1. ábra

Az ipconfig és ipconfig /all parancsok és a kapott válaszok

- **`release` vagy `renew`** – Ezek az elhagyható paraméterek csak azokon a számítógépeken működnek, amelyek az IP-címüket egy DHCP-kiszolgálótól bérlik. Az `ipconfig /release` (feloldás) parancsot beírva minden felület bérelt IP-címét visszaadjuk a DHCP-kiszolgáló(k)nak, míg ha az `ipconfig /renew` (megújítás) parancsot adjuk ki, a helyi számítógép ezzel éppen ellentétesen megpróbál kapcsolatba lépni egy DHCP-kiszolgálóval, hogy új IP-címet igényeljen tőle. Nem árt, ha tudjuk, hogy megújításkor a hálózati kártya vagy kártyák sok esetben ugyanazt az IP-címet kapják, mint korábban.

A `release` és `renew` kapcsolókat arra is használhatjuk, hogy egy több hálózati kártyát tartalmazó számítógépben egyenként oldjuk fel vagy újítsuk meg a kártyák címét. Ha a számítógép egyik csatolókártyájának a neve például `Elnk31`, ennek a kártyának a címét a következő parancsokkal oldhatjuk fel, illetve újíthatjuk meg: `ipconfig /release Elnk31`, illetve `ipconfig /renew Elnk31`.

A Mac OS X a System Preferences (Rendszerbeállítások) Network (Hálózat) alkalmazásán keresztül jeleníti meg a hálózati beállításokat (lásd a 14.2. ábrát). Mivel a Mac OS X valójában egyfajta Unix rendszer, a hálózati beállításokat úgy is kiírathatjuk, ha a Terminal (Terminál) ablakba beírjuk az `ifconfig` parancsot.

14.2. ábra

A Mac OS X Network alkalmazása lehetővé teszi a hálózati beállítások megtekintését

ARP

Az ARP (Address Resolution Protocol, névfeloldási protokoll) egy IP-címnek megfelelő fizikai (MAC) cím meghatározására használatos. A TCP/IP-hálózatokon minden állomás fenntart egy ARP-gyorsítótárat – ez egy tábla, amely az IP-címeket fizikai címeknek feleltelti meg. Az `arp` parancs lehetővé teszi, hogy megtekintsük az ARP-gyorsítótár aktuális tartalmát akár a helyi számítógépen, akár egy másikon. Az ARP-gyorsítótár frissítéséről a legtöbb esetben a protokollszoftver gondoskodik; ritkán adódik olyan helyzet, amikor az `arp` parancs segítségére van szükségünk egy hálózati kapcsolat hibaelhárításához. Mindazonáltal a parancsnak időnként hasznát vehetjük, ha az IP-címek és a fizikai címek megfeleltetésével kapcsolatban rejtélyesebb hibákat kell felderítenünk.

Az `arp` parancs azt is megengedi, hogy saját kezűleg adjuk meg a kívánt fizikai-IP címpárt. Ezt az olyan általánosan használt állomások esetében lehet érdemes megtenni, mint az alapértelmezett átjáró, illetve a helyi kiszolgálók. Ezzel a megoldással csökkenthetjük a forgalmat a hálózaton.

Az ARP-gyorsítótárban található elemek alapértelmezés szerint dinamikusak: automatikusan kerülnek a gyorsítótárba, amikor elküldünk egy irányított adatcsomagot, és az aktuális elem nem szerepel a célszámítógép gyorsítótárában. A gyorsítótár elemeinek elévülése a tárba kerülésük után azonnal megkezdődik, ezért ne lepődjünk meg, ha az ARP-gyorsítótárban csak kevés bejegyzést vagy egyet sem találunk. Elemet úgy is adhatunk a tárhoz, ha visszhangkérést intézünk egy másik számítógéphez vagy egy útválasztóhoz. A gyorsítótár elemeit az alábbi `arp` parancsokkal tekinthetjük meg:

- **`arp -a`** – Ezzel a paranccsal az összes elemet megjeleníthetjük az ARP-gyorsítótárból.
- **`arp -g`** – Ez a parancs ugyancsak az összes elemet megjeleníti az ARP-gyorsítótárból.

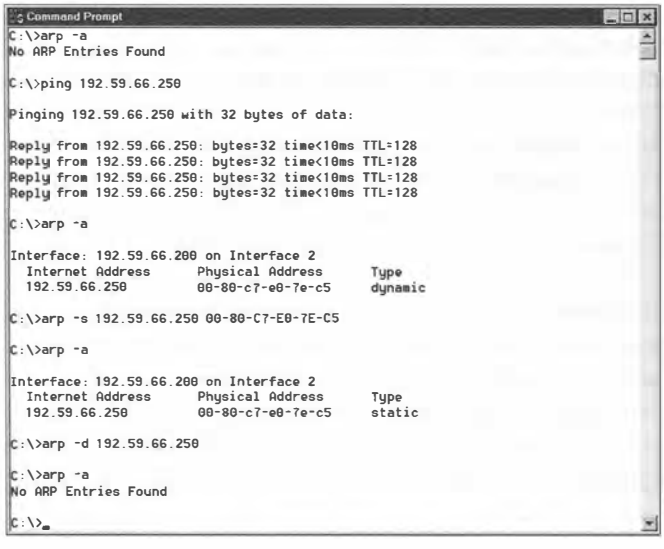
Az `arp -a` és az `arp -g` parancsot egyaránt használhatjuk. Unix rendszereken évek óta a `-g` kapcsolót használják az ARP-gyorsítótár összes bejegyzésének megjelenítéséhez. A Windows az `arp -a` parancsot használja (*a*, mint *all*, vagyis *összes*), de elfogadja a hagyományosabb `-g` kapcsolót is.

- **`arp -a <IP_cím>`** – Ha több hálózati kártyával rendelkezünk, az ARP-gyorsítótárból megtekinthetünk csak egy adott felülethez tartozó bejegyzéseket is. Ehhez az `arp -a <a felület IP-címe>` parancsot kell használnunk. Például: `arp -a 192.59.66.200`.
- **`arp -s`** – Kézi módszerrel maradandó, statikus bejegyzést is adhatunk az ARP-gyorsítótárhoz. Ez a bejegyzés a számítógép többszöri újraindítása után is hatályos marad, és automatikusan frissül, ha a saját kezűleg beállított fizikai címek használatakor hibák lépnek fel. Például ha egy kiszolgálóhoz saját kezűleg

szeretnénk bejegyzést felvenni, a `192.59.66.250` IP-címmel, illetve `0080C7E07EC5` fizikai címmel, akkor írjuk be ezt a parancsot: `arp -s 192.59.66.250 00-80-C7-E0-7E-C5`.

- **`arp -d <IP_cím>`** – Ezzel a paranccsal egy statikus elem kézi törlését hajthatjuk végre. Például: `arp -d 192.59.66.250`.

Az `arp` parancsokra és a rájuk kapott válaszokra a 14.3. ábrán láthatunk példákat.



14.3. ábra

Az arp parancsok és a kapott válaszok

Vonalproblémák

Ha egy hálózati elosztóval vagy kábellel van gond, az nem igazán a TCP/IP-vel kapcsolatos probléma, de a TCP/IP olyan diagnosztikai segédprogramjaival, mint a `ping`, a vonalproblémákat is felderíthetjük. Általánosságban, ha a hálózat korábban rendben volt, de egyszer csak nem működik, többnyire egy vonalhiba az oka. Először is, győződjünk meg róla, hogy minden hálózati kábel megfelelően van csatlakoztatva. A legtöbb hálózati kártyán, elosztón és útválasztón találhatunk jelzőfényeket, amelyek jelzik, hogy az egység be van-e kapcsolva, és készen áll-e az adatok fogadására. Az elosztók, útválasztók és kapcsolók csatlakozóhelyei ezenkívül *kapcsolatállapoti* fényekkel is rendelkeznek, amelyek mutatják, hogy az adott kapun keresztül van-e aktív hálózati kapcsolat. A hálózati kábelezés ellenőrzésére több eszköz is létezik. Ha nem rendelkezünk kábeltesztelő eszközzel, mindig megtehetjük, hogy kihúzzuk a gyanús kábelt, és egy újat dugunk a helyére, hogy lássuk, ez megoldja-e a problémát.

A vonalproblémák azonosítására a fejezet korábbi részében ismertetett `ping` parancsot is használhatjuk. Ha egy számítógép a saját címéről kap visszhangválaszt, de semmilyen más címről nem a hálózaton, a hiba a számítógépet a helyi alhálózathoz kapcsoló kábelszakaszon belül lehet.

Névfeloldási problémák

Névfeloldási probléma akkor jelentkezik, ha egy állomás neve, amelynek egy üzenetet címzünk, nem található a hálózaton. A névfeloldási problémákat egyesek nem tekintik kapcsolati problémának, mert nem feltétlenül jelentik azt, hogy a forrásszámítógép nem tud kapcsolódni a célhoz. Valójában, ahogy egy korábbi részben említettük, a névfeloldási problémák egyik leggyakoribb tünete, hogy a forrásszámítógép az IP-címével el tudja érni a célt, csak az állomásneve szerint nem. Bár egy névfeloldási probléma szigorú értelemben véve nem kapcsolati probléma, gyakorlati szempontból az, mert a mai hálózatokon az erőforrásokra állomásnévvel vagy NetBIOS-névvel hivatkoznak, és az állomásokhoz először többnyire név szerint próbálunk meg kapcsolódni. Ha ez a kísérlet nem jár sikerrel, megkezdhetjük az óra korábbi részében, a `ping` parancs bemutatásánál ismertetett hibaelhárítási lépéseket. Ha IP-cím szerint képesek vagyunk kapcsolatot létesíteni, valószínűleg a névfeloldással van gond. Sok névfeloldási probléma oka nyilvánvaló, ha végiggondoljuk a névfeloldás folyamatát (lásd a 11. fejezetben). A leggyakoribb okok között a következőket találjuk:

- A `hosts` fájl hiányzik, vagy helytelen bejegyzéseket tartalmaz.
- A névkiszolgáló nem tartózkodik a hálózaton.
- Az ügyfél beállításai hibásan hivatkoznak a névkiszolgálóra.
- Az elérni kívánt állomáshoz nem tartozik bejegyzés a névkiszolgálón.
- A parancsban használt állomásnév nem megfelelő.

Ha egy számítógéphez nem tudunk állomásnévvel kapcsolódni, próbáljunk meg kapcsolatot létesíteni egy másik számítógéppel. Ha az „A" számítógéphez képesek vagyunk kapcsolódni állomásnév szerint, de a „B" számtógéphez nem, akkor a probléma valószínűleg a „B" számítógéppel kapcsolatos, illetve azzal, ahogy a névszolgáltatás hivatkozik rá. Ha sem az „A", sem a „B" számítógép nem érhető el, akkor nagy az esély rá, hogy általánosabb probléma van a névszolgáltatási rendszerrel.

Ha olyan hálózaton botlunk névfeloldási problémákba, ahol névkiszolgáló működik, jó ötlet visszhangkérést intézni a névkiszolgálóhoz, hogy ellenőrizzük, hogy kapcsolódik-e a hálózatra. Amennyiben a névkiszolgáló a helyi alhálózaton túl található, az átjárótól kérjünk visszhangot, hogy biztosak lehessünk benne, hogy a névfeloldási kérelmek el tudják érni a névkiszolgálót. Ezenkívül ellenőrizzük még egyszer a beírt nevet, hogy az erőforrás nevét helyesen adtuk-e meg. Ha egyik említett lépés sem vezet eredményre, az `nslookup` segédprogrammal konkrét bejegyzésekről érdeklődhetünk a névkiszolgálónál. Az `nslookup` segédprogramról a 11. órán beszéltünk bővebben.

Ha olyan számítógépen dolgozunk, amelyiknek nem tudjuk az állomásnevét, használjuk a `hostname` parancsot. A `hostname` egy egyszerű parancs, amely a legtöbb operációs rendszeren elérhető, és az adott (helyi) számítógép állomásnevét adja vissza. A `hostname`-nek nincsenek kapcsolói vagy paraméterei. Egyszerűen írjuk be a `hostname` parancsot, és nézzük meg az egyetlen szóból álló választ.

Hálózati teljesítményproblémák

A hálózati teljesítményproblémák olyan problémák, amelyeknek a hatására a hálózat lassan válaszol. Mivel a TCP/IP-protokollok általában TTL-beállításokat (Time To Live, élettartam) használnak az adatcsomagok élettartamának korlátozására a hálózaton, a lassúság csomagok elvesztését és így kapcsolatok elvesztését is eredményezheti. A gyenge hálózati teljesítmény egyik szokásos oka a túlságosan nagy adatforgalom. A hálózaton azért lehet nagy az adatforgalom, mert túl sok számítógép kapcsolódik a hálózatra, vagy mert egy rosszul működő eszköz, például egy hálózati kártya szükségtelen forgalmat idéz elő a hálózaton – ez utóbbit hívják *adásviharnak* (broadcast storm). A hálózat lassúságát néha egy kiesett útválasztó okozza, amely nem továbbítja az adatokat, és így szűk keresztmetszetet hoz létre valahol a hálózaton.

A TPC/IP több segédprogramot is kínál, amelyeknek a segítségével megnézhetjük, hová tartanak az adatcsomagok, illetve statisztikákat jeleníthetünk meg a hálózat teljesítményéről. A következőkben ezeket a segédprogramokat vesszük sorra.

traceroute

A `traceroute` segédprogram az adatcsomagok útjának feltérképezésére szolgál, ahogy azok a számítógépünktől több átjárón keresztül a céljuk felé haladnak. A segédprogram által feltárt útvonal csupán egy út a forrás és a cél között – azt semmi sem garantálja vagy tételezi fel, hogy az adatcsomagok mindig ezt az útvonalat fogják követni. Ha a számítógépünk DNS használatára van beállítva, a válaszokból gyakran városok, régiók és közvetítők nevét is meghatározhatjuk. A `traceroute` parancs lassan ad eredményt: akár 10-15 másodpercet is adnunk kell neki útválasztónként.

A `traceroute` (vagy `tracert`, ha Windowst használunk) segédprogram az ICMP protokollra támaszkodva azonosítja az ügyfélszámítógép és a célszámítógép között álló útválasztókat. Az útválasztók vgy átjárók számát, amelyeken egy adatcsomag áthalad, a TTL-érték árulja el. Az eredeti kimenő ICMP Echo üzenetben használt TTL-érték módosításával a `traceroute` a következőképpen tudja megtalálni az útvonalon levő útválasztókat:

1. Egy ICMP Echo üzenetet küld a cél IP-címre, a TTL-értéket *1*-re állítva. Az első útválasztó kivon 1-et a TTL-értékből, ami azt eredményezi, hogy a TTL-érték 0-ra csökken.
2. Mivel a TTL-érték most 0, az útválasztó tudja, hogy nem szabad kísérletet tennie az adatcsomag továbbítására, ezért egyszerűen elveti azt, mert az adatcsomag élettartama lejárt. Az útválasztó ekkor egy `ICMP Time Exceeded - TTL Expired In Transit` (ICMP-időtúllépés – az élettartam átvitel közben lejárt) üzenetet küld vissza az ügyfélszámítógépnek.
3. Az ügyfélszámítógép, amely kiadta a `traceroute` parancsot, megjeleníti az említett útválasztó nevét, majd egy újabb ICMP Echo üzenetet bocsát ki, ezúttal a TTL-értéket 2-re állítva.
4. Az első útválasztó kivon 1-et a TTL-értékből, és ha tudja, továbbítja az adatcsomagot az útvonalon található következő ugrópontra. Amikor az adatcsomag eléri a második útválasztót, a TTL-érték ismét 1-gyel csökken, ami 0 értéket eredményez.
5. A második útválasztó az elsőhöz hasonlóan egyszerűen elveti a csomagot, és ugyanazt az ICMP-üzenetet küldi vissza a feladónak, mint amit az első útválasztó küldött az első alkalommal.
6. A fenti folyamat folytatódik, és a `traceroute` addig növeli, illetve az útválasztók addig csökkentik a TTL-értéket, amíg az adatcsomag végül eléri a kívánt célját.
7. Amikor a célszámítógép megkapja az ICMP Echo üzenetet, egy `ICMP Echo Reply` üzenettel válaszol rá.

Az egyes útválasztókon vagy átjárókon kívül, amelyeken az adatcsomag áthalad, a `traceroute` segédprogram azt az időt (round-trip time) is rögzíti, amennyi az egyes útválasztók eléréséhez szükséges. A megvalósítástól függően a `traceroute` egynél több Echo üzenetet is küldhet az útválasztóknak. A Windows-változatban (`tracert`) például minden útválasztó két további Echo üzenetet kap, hogy a program jobb becslést tudjon adni az elérési időre.

Mindazonáltal ebből az elérési időből nem vonhatunk le pontos következtetéseket a hálózat sebességére nézve. Sok útválasztó alacsonyabb elsőbbségi szintre helyezi az ICMP-forgalmat, és a feldolgozási idő legnagyobb részét a fontosabb adatcsomagok továbbításával tölti.

A `traceroute` parancs utasításformája egyszerűen a `traceroute` parancsból áll, amelyet egy IP-cím, egy DNS-név vagy akár egy URL követhet:

```
traceroute 198.137.240.91
traceroute www.whitehouse.gov
tracert yahoo.com (Windows rendszeren)
```

A `traceroute` és a `tracert` parancs nem csak azért hasznos, mert megmutatja, hogy milyen útvonalat jár be egy adatcsomag a célja felé – ezek a parancsok diagnosztikai képességekkel is bírnak, aminek szintén nagy hasznát vehetjük.

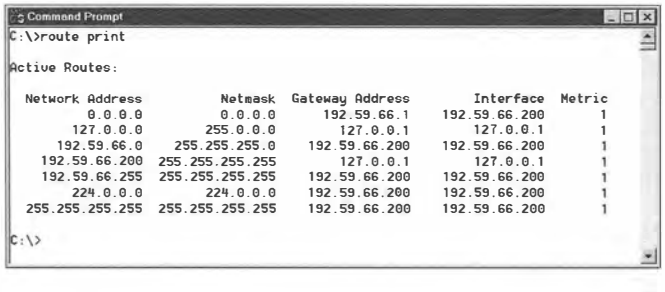
route

Ahogy a 8. órán megtanultuk, minden számítógép és útválasztó rendelkezik egy útválasztási táblázattal. A legtöbb útválasztó különleges útválasztási protokollokat használ az útválasztási információk kicserélésére, és rendszeresen, dinamikusan frissíti a tábláját. Mindazonáltal sokszor előfordul, hogy saját kezűleg kell bejegyzéseket adnunk az útválasztási táblázatokhoz az útválasztókon és gazdaszámítógépeken.

A `route` parancsot több célre is használják a TCP/IP-hálózatokban: megjeleníthetjük vele az útválasztási táblázatot, ha egy adott állomásról származó csomagok nem hatékony útvonalon haladnak. Amennyiben a `traceroute` parancs abnormális vagy nem hatékony útvonalat tár fel, a `route` segítségével meghatározhatjuk, hogy miért ez az útvonal van használatban, és esetleg beállíthatunk egy hatékonyabb útvonalat. A `route` paranccsal emellett saját kezűleg adhatunk bejegyzéseket az útválasztási táblázatokhoz, illetve törölhetjük vagy módosíthatjuk azokat. Nézzünk meg néhányat a lehetőségek közül:

- **`route print`** – A `route` parancsnak ez a formája az útválasztási táblázat aktuális bejegyzéseit jeleníti meg. A `route print` parancs kimenetére a 14.4. ábrán láthatunk egy példát. Amint megfigyelhetjük, egyes bejegyzések különféle hálózatokra hivatkoznak (ilyen például a `0.0.0.0`, a `127.0.0.0` és a `192.59.66.0`); valamint vannak adatszórási (broadcast) címek (`255.255.255.255` és `192.59.66.255`), illetve csoportos (többcímes, multicast) küldésre szolgáló címek (`224.0.0.0`). Ezek a bejegyzések automatikusan kerültek a táblázatba a hálózati csatolók IP-címeinek beállításakor.
- **`route add`** – A `route` parancsnak ezt a formáját arra használhatjuk, hogy új útválasztási bejegyzést adjunk az útválasztási táblázathoz. Például ha egy olyan útvonalat szeretnénk megadni, amely a `207.34.17.0` című, öt útválasztási ugrásnyira levő célhálózathoz vezet, és először a helyi hálózaton a `192.59.66.5` IP-címmel és a `255.255.255.224` alhálózati maszkkal rendelkező útválasztón halad át, a következő parancsot kell beírnunk:

```
route add 207.34.17.0 mask 255.255.255.224 192.59.66.5 metric 5
```



14.4. ábra

A route print parancs az útválasztási táblázatban található aktuális információkat jeleníti meg

Az így megadott útválasztási információ nem maradandó; a számítógép vagy az útválasztó újraindításakor elvész. A `route add` parancsokat ezért gyakran indítási parancsfájlokban helyezik el, hogy beállítások a számítógép vagy útválasztó újraindításakor újra életbe lépjenek.

- **`route change`** – Ezzel az utasítással egy bejegyzést módosíthatunk az útválasztási táblázatban. Az alábbi parancs például az adatokat egy másik útválasztóhoz irányítja, amely közvetlenebb, csupán három ugrásból álló útvonalat használ a célig:

```
route change 207.34.17.0 mask 255.255.255.224 192.59.66.7 metric 3
```

- **`route delete`** – Ezzel a paranccsal egy bejegyzést törölhetünk az útválasztási táblázatból:

```
route delete 207.34.17.0
```

netstat

A `netstat` segédprogram az IP, TCP, UDP és ICMP protokollokkal kapcsolatos statisztikákat jelenít meg. Ezek a statisztikák olyan értékeket mutatnak, mint az elküldött, illetve fogadott adatcsomagok száma, illetve az esetleg fellépett különféle hibák.

Ne lepődjünk meg, ha a számítógépünk időnként olyan adatcsomagokat fogad, amelyek hibát, adatelvetést vagy összeállítási kudarcot okoznak. A TCP/IP jól tűri az ilyen jellegű hibákat, és önműködően újraküldi az adatcsomagot. Adatelvetésre akkor kerül sor, ha egy adatcsomagot téves helyre kézbesítenek. Amennyiben a számítógépünk útválasztóként működik, akkor is elveti az adatcsomagokat, ha a TTL-értékük eléri a nullát. Összeállítási kudarc akkor következhet be, ha nem érkezik meg minden töredék azon az időn belül, amelyet a beérkezett töredékek TTL-értéke határoz meg. A hibákhoz és az adatelvetéshez hasonlóan az egyszer-egyszer fellépő összeállítási kudarcok miatt sem kell aggódnunk. Mindhárom esetben akkor lehet okunk a hiba forrásának megkeresésére, ha a hibák összesítve az összes fogadott IP-csomagnak már jelentős százalékát teszik ki, vagy ha a számuk gyorsan nő.

A `netstat` parancs különféle kapcsolóit az alábbiakban ismertetjük:

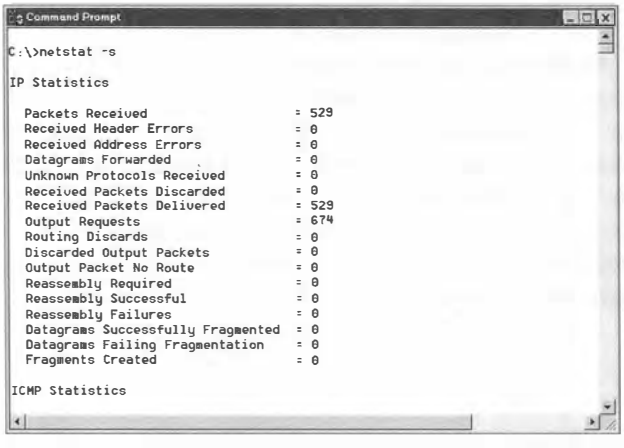
- **`netstat -s`** – Ezzel a kapcsolóval protokollonként jeleníthetjük meg a statisztikákat. Ha az olyan felhasználói alkalmazások, mint a webböngésző, szokatlanul lassan működnek, vagy nem képesek olyan adatokat megjeleníteni, mint a weboldalak, érdemes lehet ezzel a paranccsal információt kérni. A statisztikák soraiban olyan szavakat kell keresnünk, mint az `error` (hiba), a `discard` (adatelvetés) vagy a `failure` (összeállítási kudarc). Ha az ilyen sorokban található számok a fogadott IP-csomagok jelentős százalékát teszik ki, további vizsgálatokra lehet szükség.

- **netstat -e** – Ez a kapcsoló az Ethernet-hálózatról szolgáltat statisztikákat. A felsorolt elemek között olyanokat találunk, mint a teljes bájt-, hiba- és adatelvetésszám, az irányított adatcsomagok száma, valamint az adatszórások száma. A statisztikák mind az elküldött, mind a fogadott adatcsomagokra vonatkozó értékeket megmutatják.
- **netstat -r** – Ez a kapcsoló a `route print` parancshoz hasonlóan az útválasztási táblázatban található információk megjelenítésére szolgál. Az aktív útvonalakon kívül az éppen aktív kapcsolatokat is megmutatja.
- **netstat -a** – Ezzel a kapcsolóval az összes aktív kapcsolatot soroltathatjuk fel, beleértve a létrejött kapcsolatokat és amelyek kapcsolati kérelemre várnak.

A következő három kapcsoló az `-a` kapcsolóval megjelenített információk egy-egy részhalmazát jeleníti meg:

- **netstat -n** – Ez a kapcsoló az összes létrejött aktív kapcsolatot mutatja meg.
- **netstat -p TCP** – Ezzel a kapcsolóval a létrejött TCP-kapcsolatokat jeleníthetjük meg.
- **netstat -p UDP** – Ez a kapcsoló a létrejött UDP-kapcsolatok megjelenítésére szolgál.

A `netstat -s` paranccsal megjelenített statisztikákra a 14.5. ábra mutat egy példát.



14.5. ábra
Protokollonkénti statisztikák megjelenítése a netstat paranccsal

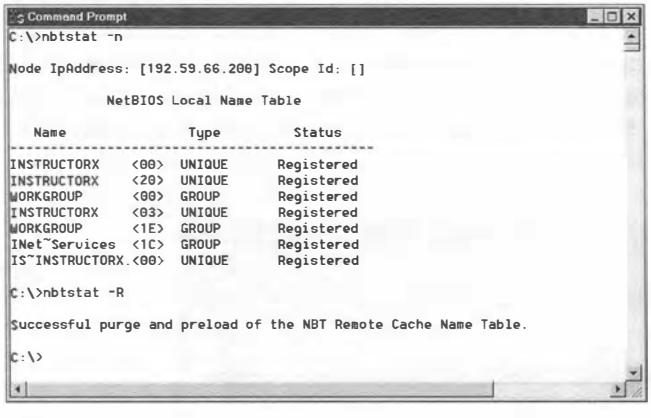
Nbtstat

Ahogy a 11. órán megtanultuk, a NetBIOS egy névfeloldási rendszer, amelyet számos régi Windows-hálózaton használnak. Az `nbtstat` (NetBIOS TCP/IP felett statisztikák) segédprogram a TCP/IP feletti NetBIOS-ról szolgáltat statisztikákat. A parancs lehetővé teszi, hogy megtekintsük a helyi vagy egy távoli számítógépen található NetBIOS-névtáblázat tartalmát.

A parancs következő kapcsolói a helyi számítógépen használatosak:

- **nbtstat -a** – Ezzel a paranccsal kiürítjük és újratöltjük a NetBIOS-névgyorsítótárat. Erre akkor lehet szükség, ha be szeretnénk tölteni az LMHosts fájlba újonnan felvett bejegyzéseket. (Az LMHosts-bejegyzésekkel a 11. órán foglalkoztunk.)
- **nbtstat -n** – Ez a parancs a helyi számítógépen bejegyzett neveket és szolgáltatásokat sorolja fel.
- **nbtstat -c** – Ezzel a paranccsal a NetBIOS-névgyorsítótár tartalmát jeleníthetjük meg, amely azoknak a számítógépeknek az IP-címeihez tárolja a megfelelő NetBIOS-neveket, amelyekkel a számítógépünk nemrégiben kommunikált.
- **nbtstat -r** – Ezzel a paranccsal más számítógépek bejegyzéseit és feloldott neveit írathatjuk ki, illetve hogy a bejegyzésük és a feloldásuk adatszórás útján vagy egy névkiszolgáló által történt-e.

Az `nbtstat` parancsok kimenetére a 14.6. ábrán láthatunk példát.



14.6. ábra
Az nbtstat parancsok és a kapott válaszok

Az `nbtstat` parancs távoli számítógépek NetBIOS-névtáblázatának megtekintésére is használható; a kimenet hasonló a helyi számítógépen kiadott `nbtstat -n` parancséhoz.

- **nbtstat -A <*IP_cím*>** – Egy másik számítógép névtáblázatát jeleníti meg a fizikai címekkel együtt, az említett számítógép IP-címét használva.
- **nbtstat -n <*NeTBIOS_név*>** – Egy másik számítógép névtáblázatát jeleníti meg a fizikai címekkel együtt, az említett számítógép NetBIOS-nevét használva.

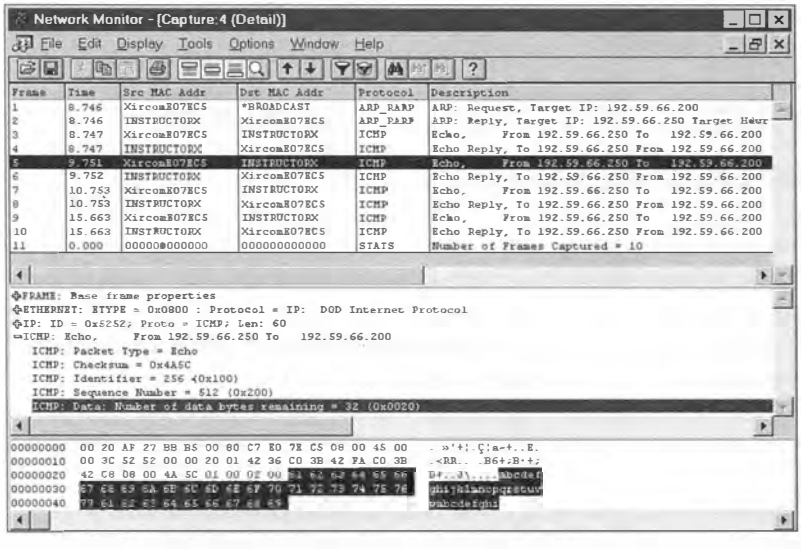
Ehhez hasonlóan az `nbtstat` parancsnak van két másik kapcsolója is, amelyekkel egy távoli számítógép által megnyitott NetBIOS-kapcsolatok listáját tekinthetjük meg (ezt a listát hívják kapcsolati táblázatnak):

- **nbtstat -S <*IP_cím*>** – Egy másik számítógép NetBIOS-kapcsolati táblázatát jeleníti meg az említett számítógép IP-címét használva.
- **nbtstat -s <*NeTBIOS_név*>** – Egy másik számítógép NetBIOS-kapcsolati táblázatát jeleníti meg az említett számítógép NetBIOS-nevét használva.

Csomaglehallgatók

A *csomaglehallgatóként* (sniffer, „szimatoló") ismert segédprogramok egy átmeneti tárba vagy fájlba rögzítenek adatokat a hálózatról. Miután az adatokat elfogtuk, keretenként vagy csomagonként jeleníthetjük meg az adattartalmat. A csomaglehallgatók a hálózati forgalom rejtélyesebb hibáinak elemzésében lehetnek a hasznunkra, de arra is használhatjuk őket, hogy megkeressük a forrását azoknak a sérült csomagoknak, amelyek esetleg egy hibásan működő eszközről származnak. Egy Ethernet-keretet a fizikai címe alapján követhetünk vissza. Az okok felderítése érdekében bármely protokollszintről elemezhetünk fejlécinformációkat (lásd a 3., 4. és 6. órát).

A 14.7. ábra tíz adatcsomag sorozatát mutatja, amelyeket egy `ping` parancs kiadásával küldtünk el. A felső ablakban a tíz adatcsomag látható, egy ARP-kérelemmel és egy ARP-válasszal kezdve, amelyet négy ICMP kérelem–válasz pár követ. A középső ablak a visszafejtett ICMP-fejlécet mutatja, az alsó keretben pedig az adatcsomagban található 32 bájtnyi adatot tekinthetjük meg. A 32 adatbájt a teljes ábécéből, majd az `abcdef` betűkből áll.



14.7. ábra

Az adatforgalom nézete egy ping parancs kiadása után

Hibaelhárítás a kapcsolati segédprogramok segítségével

A protokollverem különböző szintjein működő alkalmazásokat kipróbálva gyakran pontosan meghatározhatjuk, hogy a verem melyik összetevője okozza a problémát. Ahogy a `ping` segédprogram ismertetésekor említettük, a hálózati problémák elhárításakor van egy sorrend, amit követnünk kell. Hibaelhárításnál általában egyszerű, alapvető parancsokkal kezdjük, és ha ezek a várt eredményt adják, erre a tudásra építve már továbbléphetünk az egyre több hálózati szolgáltatást igénybe vevő parancsok felé. Egy hálózati probléma elhárításához tehát kövessük az alábbi lépéseket:

1. Kezdjük az `ifconfig` és `ipconfig` paranccsal, illetve valamilyen ezekhez hasonló eszközzel, hogy biztosan tudjuk az aktuális IP-címet, alhálózati maszkot, illetve az alapértelmezett átjáró címét.
2. Lépjünk tovább a `ping` parancsokra, és adjuk ki őket a korábban ismertetett sorrendben. Ha a `ping` parancsok a várt eredményt adják, bízhatunk benne, hogy a két alsó réteg, így a hálózati csatolókártya és a hálózati kábelezés rendben van.
3. Egy webböngészővel vagy hasonló alkalmazással kapcsolódjunk egy webkiszolgálóhoz. Ha ez sikerül, tudhatjuk, hogy a TCP és a csatolófelület működik. Ha nem, próbáljunk ki egy másik alkalmazást, amely a TCP-t és a csatolókat (socket) használja, például egy FTP-ügyfélprogramot. Ha az sem működik, valószínűleg a TCP-re vagy a csatolókra kell összpontosítanunk, ha meg akarjuk találni a probléma forrását.

FTP

Az *FTP* (*File Transfer Protocol, fájlátviteli protokoll*) széles körben használatos protokoll, amely lehetővé teszi, hogy fájlokat vigyünk át két számítógép között egy TCP/IP-hálózaton. A fájlátviteli alkalmazások (amelyeknek jellemzően szintén `ftp` a neve) az FTP protokollt használják a fájlok átvitelére. Az átvitel úgy működik, hogy a felhasználó egy FTP-ügyfélprogramot futtat a számítógépén, a másik gépen pedig egy olyan FTP-kiszolgálóprogram fut, mint Unix/Linux gépeken az `ftpd` (FTP-démon), más platformokon pedig egy FTP-szolgáltatás. Sok FTP-ügyfélprogram parancssoros, de grafikus változatok is rendelkezésre állnak. Az FTP-t elsősorban fájlok átvitelére használják, bár más feladatokat, például könyvtárak létrehozását és eltávolítását, illetve a fájltartalom kiíratását is képes végrehajtani.

A Unix világában a *démon* egy folyamat, amely a háttérben fut, és kérésre valamilyen szolgáltatást nyújt. A démonokat a Windows világában *szolgáltatásoknak* hívják.

Az FTP a TCP protokollra támaszkodik, ezért egy megbízható, kapcsolatközpontú munkameneten keresztül működik az ügyfél- és a kiszolgálógép között. A szabványos FTP-démon (a kiszolgálón) a 21-es TCP-kaput figyeli az ügyféltől érkező kérelmekre várva. Amikor az ügyfél kérelmet küld, egy TCP-kapcsolat kezdeményezésére kerül sor (lásd a 6. fejezetet). A távoli felhasználót ezt követően az FTP-kiszolgáló azonosítja, és a munkamenet megkezdődik. A klasszikus, szöveg alapú FTP-munkameneteknél a távoli felhasználónak parancssoros felületen keresztül kell kapcsolatot tartania a kiszogálóval. Az általánosan használt parancsok feladata az FTP-munkamenet elindítása és leállítása, a mozgás a távoli környvtárszerkezetben, valamint a fájlok fel- és letöltése. Az újabb, grafikus FTP-ügyfélprogramokban a parancssor helyett egy grafikus felületen mozoghatunk a könyvtárak között, illetve mozgathatjuk a fájlokat.

Az FTP-t a Weben is széles körben használják, és az FTP protokollt be is építették a legtöbb webböngészőbe. Amikor fájlokat töltünk le egy webböngészőn keresztül, néha észrevehetjük, hogy a címsávban szereplő URL az `ftp://` előtaggal kezdődik.

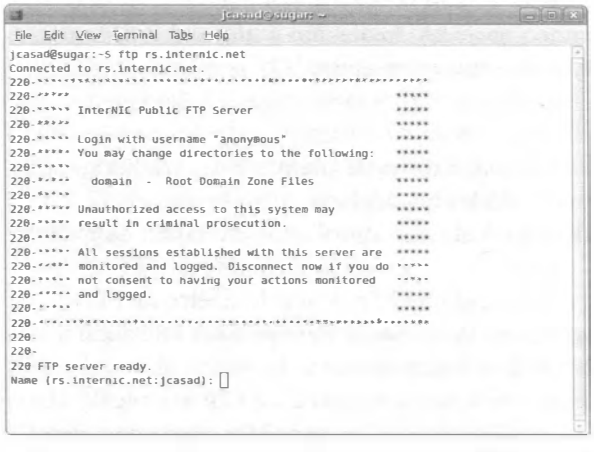
A legtöbb számítógépen úgy indíthatunk el egy szöveges alapú FTP-munkamenetet, hogy beírjuk az `ftp` parancsot, majd utána az FTP-kiszolgáló állomásnevét vagy IP-címét. Az FTP ez után elkéri a felhasználói azonosítónkat és a jelszavunkat, amelyek segítségével az FTP-kiszolgáló megállapítja, hogy jogosult felhasználók vagyunk-e, és milyen jogokkal rendelkezünk. Előfordulhat például, hogy a felhasználói fiókunk csak olvasási joggal rendelkezik, de az is lehet, hogy olvasási és írási műveletekre egyaránt jogosultak vagyunk. Sok FTP-kiszolgáló nyilvános használatú, így megengedi, hogy az `anonymous` (névtelen) felhasználóazonosítóval lépjünk be. Ha felhasználóazonosítóként az `anonymous` fióknevet használjuk, lényegében bármilyen jelszót beírhatunk, de általában az e-mail címet szokás megadni. Ha egy FTP-kiszolgálót nem a nagy nyilvánosságnak szántak, úgy állítják be, hogy ne engedje meg a névtelen hozzáférést. Ilyen esetben egy létező felhasználónevet és jelszót kell megadnunk ahhoz, hogy hozzáférést kapjunk. Ezeket általában az FTP-kiszolgáló rendszergazdája állítja be és osztja ki.

Sok FTP-ügyfélmegvalósítás Unix- és DOS-parancsok használatát egyaránt lehetővé teszi. A ténylegesen beírandó parancsok a használt ügyfélszoftvertől függnek. Amikor az FTP segítségével viszünk át fájlokat, meg kell adnunk az átvinni kívánt fájl típusát – ez a leggyakrabban „binary" (bináris) vagy „ASCII" lehet. Az ASCII típust akkor kell választanunk, ha egyszerű szövegfájlt szeretnénk átvinni, míg a binary típust akkor, ha programfájlt, szövegszerkesztőben készített dokumentumot vagy képfájlt töltünk le vagy fel. Az alapértelmezett fájlátviteli mód az ASCII.

Nem árt, ha tudjuk, hogy sok FTP-kiszolgáló Unix vagy Linux rendszerű számítógépen működik. Mivel ez a két rendszer megkülönbözteti a kis- és nagybetűket, a fájlnevek beírásakor ügyelnünk kell rá, hogy a betűállásnak megfelelően adjuk meg a neveket. Alapértelmezés szerint a fájlok le- és feltöltési helye a helyi számítógépnek az a könyvtára, ahonnan az FTP-munkamenetet kezdeményezzük.

Az alábbiakban a leggyakrabban használt FTP-parancsokat mutatjuk be:

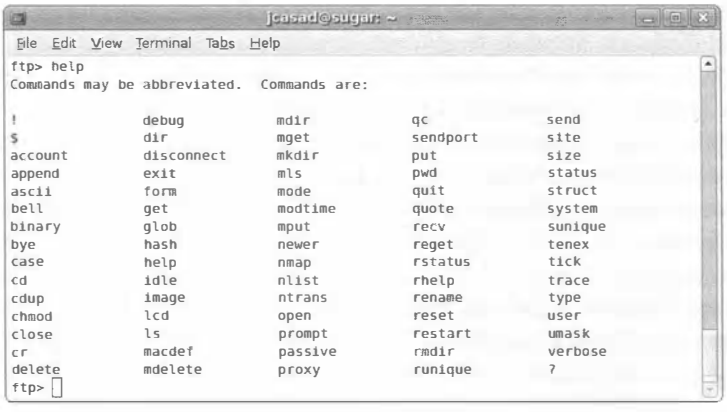
- **ftp** – Az `ftp` parancs az FTP-ügyfélprogram elindítására szolgál. Az `ftp` parancsot beírhatjuk önmagában, de megadhatunk utána egy IP-címet vagy tartománynevet is. A 14.8. ábrán azt láthatjuk, hogy az `ftp rs.internic.net` parancs beírásával az `rs.internic.net` címmel kezdeményeztünk FTP-munkamenetet. Amint láthatjuk, rengeteg információ érkezik válaszul.
  Az első sor azt jelzi, hogy kapcsolatban vagyunk. A `220`-szal kezdődő sorok mind egy testreszabott bejelentkezési üzenet részei, ami minden felhasználó számára megjelenik. A következő sor a felhasználói azonosítót kérné, de a bejelentkezési üzenetből látszik, hogy névtelen hozzáférést kaptunk. Ha az e-mail címünket kellene jelszóként megadnunk, egy `331`-gyel kezdődő testreszabott rendszerüzenet jelenne meg (a rendszerüzeneteket mindig egy szám előzi meg). A jelszó beíráskor nem jelenik meg a képernyőn.



14.8. ábra
FTP-munkamenet indítása

- **user** – A `user` parancs az adott munkamenethez tartozó felhasználóazonosító és jelszó megváltoztatására szolgál. Az új felhasználóazonosítót és jelszót az FTP ugyanúgy kéri, mint amikor az `ftp` parancsot használjuk. Ez a parancs lényegében megegyezik azzal, mintha kilépnénk az `ftp`-ből, és új felhasználóként elölről kezdenénk a munkamenetet.
- **help** – A `help` parancs az FTP-ügyfélprogramunkban elérhető FTP-parancsokat jeleníti meg (lásd a 14.9. ábrát).
- **ls vagy dir** – Az `ls` vagy `ls -l` Unix/Linux-paranccsal, illetve a `dir` Windows-paranccsal egy könyvtár tartalmát írathatjuk ki. Válaszként az FTP-kiszolgáló aktuális munkakönyvtárában található fájl- és könyvtárnevek listáját kapjuk meg. A tényleges könyvtártartalom-lista két rendszerüzenet (a `150` és `226` számokkal jelölt sorok) között található, és az aktuális munkakönyvtár összes fájlját és

alkönyvtárát tartalmazza. Az `ls -l` parancs ugyanúgy működik, mint az `ls` parancs, de további részleteket is közöl, például az olvasási és írási engedélyeket, illetve a fájlok létrehozási idejét.



14.9. ábra

Az FTP-parancsok megjelenítéséhez írjuk be a help parancsot az FTP-program készenléti jelénél

- **`pwd`** – A `pwd` parancs az aktuális munkakönyvtár (nem a helyi számítógépen, hanem a távoli kiszolgálón található könyvtár) nevét írja ki.
- **`cd`** – A `cd` parancs az FTP-kiszolgáló aktuális munkakönyvtárának megváltoztatására szolgál.
- **`mkdir`** – A Unix/Linux `mkdir` parancsa egy könyvtárat hoz létre az FTP-kiszolgálón, az aktuális munkakönyvtáron belül. Ez a parancs általában nem engedélyezett a névtelen FTP-munkamenetekben.
- **`rmdir`** – Az `rmdir` Unix-parancs töröl egy könyvtárat az FTP-kiszolgáló aktuális munkakönyvtárából. Ez a parancs általában nem engedélyezett a névtelen FTP-munkamenetekben.
- **`binary`** – A `binary` parancs az FTP-ügyfélprogramot bináris átviteli módra állítja az alapértelmezett ASCII átviteli módról. A bináris módra akkor lehet szükség, ha a `get`, `put`, `mget` és `mput` parancsok segítségével bináris fájlokat, például programokat és képeket szeretnénk átvinni.
- **`ascii`** – Az `ascii` parancs az FTP-ügyfélprogramot bináris módból ASCII átviteli módba kapcsolja.
- **`type`** – A `type` parancs az éppen használatban levő fájlátviteli módot (ASCII vagy bináris) jeleníti meg.
- **`status`** – A `status` parancs az FTP-ügyfélprogram különböző beállításairól jelenít meg információkat. Ilyen információ például, hogy az ügyfél milyen módban (bináris vagy ASCII) működik, illetve hogy az ügyfél bővebb rendszerüzenetek megjelenítésére van-e beállítva.

- **get** – A `get` parancs letölt egy fájlt egy FTP-kiszolgálóról az FTP-ügyfélre. Ha a `get` parancs után egyetlen fájlnevet írunk, a fájl az FTP-kiszolgálóról az FTP-ügyfél munkakönyvtárába másolódik. Ha két fájlnevet adunk meg, a második név az ügyfélen újonnan létrehozott fájl nevét fogja megadni. Ha elhagyjuk a második fájlnevet, az FTP-program általában kér egyet.
- **mget** – Az `mget` parancs hasonlóan működik, mint a `get`, de több fájl átvitelét is lehetővé teszi.
- **put** – A `put` parancs feltölt egy fájlt az FTP-ügyfélről az FTP-kiszolgálóra. Ha a `put` parancs után egyetlen fájlnevet írunk, a fájl az FTP-ügyfélről az FTP-kiszolgálóra másolódik. Ha két fájlnevet adunk meg a `put` után, a második név a kiszolgálón újonnan létrehozott fájl nevét fogja megadni. Ha elhagyjuk a második fájlnevet, az FTP-program általában kér egyet.
- **mput** – Az `mput` parancs ugyanazt csinálja, mint a `put`, de egyetlen paranccsal több fájl átvitelét is lehetővé teszi.
- **open** – Az `open` paranccsal új munkamenetet indíthatunk egy FTP-kiszolgálóval. Ez lényegében egy rövidebb út arra, hogy kilépjünk az FTP-programból, és elölről kezdjük a műveletet. Az `open` parancs segítségével teljesen új FTP-kiszolgálón is nyithatunk munkamenetet, de a korábbi kiszolgálóval is kezdeményezhetjük a munkamenet újbóli megnyitását.
- **close** – A `close` parancs befejezi az aktuális munkamenetet az FTP-kiszolgálóval. Az FTP-ügyfélprogram nyitva marad, és az `open` paranccsal új munkamenetet kezdeményezhetünk a kiszolgálóval.
- **bye vagy quit** – Ezek a parancsok bezárják az FTP-munkamenetet, és kilépnek az FTP-ügyfélprogramból is.

Bár a fenti felsorolás nem tartalmaz minden FTP-parancsot, az FTP-munkamenetek során leggyakrabban alkalmazott parancsokról jó képet ad.

A mai számítógéprendszerek többsége támogatja a parancssoros fájlátvitelt, a grafikus felületű FTP-ügyfélprogramok új nemzedéke azonban szükségtelenné teszi a parancssori bevitelt. Azok a felhasználók, akik sűrűn használják az FTP-t, gyakran inkább egy olyan grafikus ügyfélprogrammal dolgoznak, amely a fájlokat a szokványos fájlkezelő („intéző") programokhoz hasonlóan jeleníti meg és kezeli.

Az FTP viszonylag régi protokoll, amelyet még az előtt fejlesztettek ki, hogy a biztonságos hálózatok nagyobb figyelmet kaptak volna. A szabvány későbbi átdolgozásai, például az RFC 2228 (FTP Security Extensions, biztonsági bővítmények az FTP-hez), fontos védelmi intézkedéseket határoztak meg, például biztonságosabb hitelesítést, de az FTP-t ma sem tartják túl biztonságosnak.

A biztonsággal kapcsolatos aggodalmak ellenére az FTP továbbra is meglehetősen népszerű. Az FTP protokoll kényelmes módszert biztosít a szokványos dokumentumok és az olyan fájlok fel- és letöltésére, amelyek túl nagyok ahhoz, hogy elektronikus levél-

ben továbbítsuk őket. A dokumentumok FTP-n keresztüli feltöltésének egyik előnye a levélben történő elküldéssel szemben, hogy FTP-parancsokkal ellenőrizhetjük a fájl jelenlétét a kiszolgálón, és így megbizonyosodhatunk róla, hogy a fájl célba ért.

Akik az FTP-nél biztonságosabb módszerre vágynak, az *SFTP* (Secure File Transfer Protocol, biztonságos fájlátviteli protokoll) protokollt és a hozzá tartozó programot használhatják, amely az FTP-hez hasonló szolgáltatásokat nyújt, de titkosított hálózati kapcsolaton keresztül. Az SFTP valójában FTP a titkosított SSH szállítási (átviteli) protokoll felett. (Az SSH-ról és az egyéb titkosítási módszerekről a 15. és 23. órán beszélünk bővebben.)

Az SFTP fokozatosan felváltja az FTP-t az olyan helyzetekben, ahol nagyobb biztonságra van szükség, ugyanakkor az FTP-elérés – beleértve a névtelen FTP-elérést, amihez semmilyen biztonsági rendszer nem szükséges – messzire visszanyúló hagyománya biztosítja, hogy az FTP továbbra is fontos szerepet fog játszani az internetes kommunikációban.

Bár a klasszikus FTP protokoll nem tesz lehetővé titkosított kommunikációt, az FTP-t is használhatjuk titkosított kapcsolaton keresztül. Egy virtuális magánhálózaton (VPN, lásd a 23. fejezetben) keresztül működő FTP-ügyfélprogram például ugyanolyan biztonságos, mint az SFTP. Az SFTP használata ugyanakkor általában kényelmesebb, mert a titkosítás részleteiről automatikusan gondoskodik.

TFTP

A *TFTP* (*Trivial File Transfer Protocol, egyszerű fájlátviteli protokoll*) protokollt fájlok átvitelére használják egy TFTP-ügyfél és egy TFTP-kiszolgáló, vagyis a `tftpd` démont futtató számítógép között. Ez a protokoll az ávitelhez az UDP-t használja, és az FTP-től eltérően nem követeli meg a felhasználótól, hogy bejelentkezzen, ha fájlokat szeretne átvinni. Mivel a TFTP-hez nem szükséges felhasználói bejelentkezés, általában biztonsági résnek tekintik, különösen ha a TFTP-kiszolgáló megengedi az írást is.

A TFTP protokollt úgy tervezték, hogy elég kicsi legyen ahhoz, hogy az UDP protokollal együtt meg lehessen valósítani egy PROM- (Programmable Read Only Memory, programozható írásvédett memória) lapkán. A TFTP protokoll képességei az FTP-vel összehasonlítva korlátozottak (erre utal a nevében a *trivial*): csak fájlok olvasására és írására képes; könyvtárak létrehozására és törlésére, a könyvtárak tartalmának kiíratására, illetve a felhasználó beléptetésére nem. A TFTP protokollt elsősorban az RARP és a BOOTP protokollokkal együtt, lemez nélküli munkaállomások elindítására, valamint egyes esetekben új rendszerkódok vagy foltok útválasztókra és más hálózati eszközökre történő feltöltésére használják. A TFTP-vel a *netascii* néven ismert ASCII-formátumban, illetve az *octet* nevű bináris formátumban vihünk át fájlokat; a harmadik, *mail* nevű formátumot már nem használják.

Amikor a felhasználó beír egy TFTP-utasítást a parancssorba, a számítógép kapcsolatot létesít a kiszolgálóval, és végrehajtja a fájlátvitelt. Az átvitel befejeztével a munkamenet bezárul, és végérvényesen véget ér. A TFTP-utasítások alakja a következő:

```
TFTP [-i] host [get | put] <forrásfájl_neve> [<célfájl_neve>]
```

Ha többet szeretnénk tudni a TFTP protokollról, olvassuk el az RFC 1350-et.

Távmásolás

Az `rcp` (remote copy, távmásolás) parancs az `ftp` alternatívájának tekinthető: fájlok másolását teszi lehetővé a hálózaton keresztül. Az `rcp` a Unix `cp` (copy, másolás) parancsának távoli változata, amelynek a használatakor nem kell felhasználói azonosítót vagy jelszót megadnunk. Némi felületes biztonságot csak az ad, hogy a számítógépünk nevének szerepelnie kell két kiszolgálói fájl, az `rhosts`, illetve a `hosts.equiv` valamelyikében. Az `rcp` parancs egy helyi számítógép és egy gazdakiszolgáló, illetve két távoli számítógép között teszi lehetővé a fájlok másolását a felhasználóknak. A parancs utasításformája a következő:

```
rcp [állomásnév1]:fájlnév1 [állomásnév2]:fájlnév2
```

- ***állomásnév1*** – A forrásszámítógép állomásnevét vagy teljesen minősített tartománynevét (FQDN, Fully Qualified Domain Name) adja meg (nem kötelező). Ezt az állomásnevet akkor kell megadnunk, ha a forrásfájl egy távoli számítógépen található. Az állomásnevekről és az FQDN-ekről a 11. órán beszéltünk bővebben.
- ***fájlnév1*** – A forrásfájl elérési útját és fájlnevét adja meg.
- ***állomásnév2***– A célszámítógép állomásnevét vagy teljesen minősített tartománynevét adja meg (nem kötelező). Ezt az állomásnevet akkor kell megadnunk, ha a célfájl egy távoli számítógépen található.
- ***fájlnév2*** – A célfájl elérési útját és fájlnevét adja meg.

A következőkben néhány példát mutatunk az `rcp` parancs használatára. Az első egy távoli Unix-számítógépről másol egy fájlt a helyi állomásra:

```
rcp server3.corporate.earthquakes.txt earthquakes.txt
```

Ez a parancs pedig a helyi gépről másol egy fájlt egy távoli számítógépre:

```
rcp earthquakes.txt server3.corporate.earthquakes.txt
```

Az `rcp` parancs segítségével két távoli állomás között is másolhatunk fájlokat. A távmásolásról és az egyéb távoli hozzáférési lehetőségekről a 15. órán beszélünk majd bővebben. Az `rcp` népszerűsége az utóbbi években a biztonsági hiányosságai miatt megkopott, és a helyét az `scp` (secure copy, biztonságos másolás) nevű új program vette át, amely ugyanazokra a műveletekre képes, mint az `rcp`, de titkosított kapcsolaton keresztül működik. Az `scp` az SSH programcsomag része, amelyről a 15. órán fogunk tanulni.

A hálózati fájlhozzáférés beépítése

Az olyan segédprogramok, mint az `ftp` vgy a `tftp`, önálló alkalmazások, amelyek a TCP/IP-protokollverem alkalmazásrétegében működnek. Ezek a segédprogramok a megjelenésükkor nagy előrelépésnek számítottak, és bizonyos esetekben még ma is hasznosak, de a gyártók és azok, akik az Internet jövőjét alakítják, sokoldalúbb megoldásokat kezdtek keresni. A céljuk az, hogy a távoli fájlhozzáférést zökkenőmentesen egybeépítsék a helyi fájlhozzáféréssel, hogy a helyi és a távoli erőforrások együtt, közös felületen jelenjenek meg.

Ahogy a 7. órán megtanultuk, ez az egyesített hálózati fájlhozzáférés egy *átirányítót* (vagy *kérelmezőt*) igényel az ügyfélszámítógépen, amely értelmezi az erőforrás-kérelmeket, és a hálózathoz intézett kérelmeket a hálózatra irányítja. A megoldás másik részét egy általános célú fájlhozzáférési protokoll képezi, amely egy teljes protokollréteget hoz létre, amelyen keresztül a grafikus felhasználói felületű eszközök és más alkalmazások elérhetik a hálózatot. A helyi hálózatokon ma már ezt a fájlhozzáférési módszert részesítik előnyben. A következőkben egy olyan protokollpárt mutatunk be, amely egyesített hálózati fájlhozzáférést kínál:

- **NFS (Network File System, hálózati fájlrendszer)** – Unix és Linux rendszereken használatos protokoll.
- **SMB (Server Message Block, kiszolgálói üzenetblokk)** – Windows-ügyfelek számára távoli fájlhozzáférést nyújtó protokoll.

Ezek a protokollok jól mutatják a TCP/IP alkalmazásrétegének erejét, és annak az előnyeit, ha egy hálózati rendszert egy jól meghatározott protokollverem köré építünk, amelyben az alacsonyabb szintű protokollok alapot biztosítanak a felettük levő, szűkebb feladatkörrel rendelkező protokolloknak.

NFS

Az NFS-t (Network File System, hálózati fájlrendszer) eredetileg a Sun cég fejlesztette ki, de ma már a Unix, a Linux és sok más rendszer is támogatja. Az NFS lehetővé teszi a felhasználóknak, hogy távoli számítógépeken található fájlokat és könyvtárakat érjenek el (olvassanak, írjanak, hozzanak létre, illetve töröljenek), ugyanúgy, mintha azok a helyi számítógépen lennének. Mivel az NFS-t úgy tervezték, hogy láthatatlan felületet nyújtson a helyi és a távoli fájlrendszerek között, és mivel a megvalósítása mindkét számítógép operációs rendszerén belül megtalálható, semmilyen változtatást nem igényel az alkalmazásokban. A programok az NFS-en keresztül újrafordítás vagy más módosítások szükségessége nélkül képesek egyaránt elérni helyi és távoli fájlokat. A felhasználó számára minden fájl és könyvtár úgy jelenik meg és viselkedik, mintha csak a helyi fájlrendszeren lenne.

Az NFS eredeti megvalósítása az UDP protokollt használta az ávitelhez, és helyi hálózatokon (LAN) való használatra szánták. A későbbi átdolgozások azonban megengedték a TCP protokoll használatát is, amelynek nagyobb megbízhatósága kibővítette az NFS lehetőségeit, így az NFS-t most már nagy kiterjedésű hálózatokon (WAN) is igénybe vehetjük.

Az NFS-t úgy tervezték, hogy független legyen az operációs rendszerektől, szállítási (átviteli) protokolloktól és a hálózat fizikai felépítésétől, hogy az NFS-ügyfelek bármilyen NFS-kiszolgálóval képesek legyenek együttműködni. Ezt a függetlenséget az ügyfél- és a kiszolgálógép közötti távoli eljáráshívások (RPC, Remote Procedure Call) biztosítják. A távoli eljáráshívás olyan művelet, amely lehetővé teszi az egyik számítógépen futó programnak, hogy a másik számítógépen futó programból hívjon meg kódrészeket. Az RPC már sok éve létezik, és számos operációs rendszer támogatja. Az NFS esetében az ügyfél operációs rendszere bocsátja ki a távoli eljáráshívásokat a kiszolgálón található operációs rendszer felé.

Mielőtt az NFS rendszeren használatba vehetnénk a távoli fájlokat és könyvtárakat, először *csatlakoztatnunk* (mount) kell azokat. Csatlakoztatás (befűzés) után a távoli fájlok és könyvtárak ugyanúgy jelennek meg és viselkednek, mintha a helyi fájlrendszeren lennének.

Az NFS protokoll legfrissebb változata a 4-es, amelyet az RFC 3530 ír le. Az NFS korábbi változatairól az RFC 1094-ből és az RFC 1813-ból tudhatunk meg többet. Az NFS megvalósítása operációs rendszerenként más és más lehet. Azt, hogy miként állíthatjuk be az NFS-t az operációs rendszerünkön, a gyártótól kapott dokumentációból tudhatjuk meg.

SMB

Az *SMB* (Server Message Block, kiszolgálói üzenetblokk) a Windows felhasználói felületének hálózati működésre képes eszközeit (Intéző, Hálózati helyek, Hálózati meghajtó csatlakoztatása) támogató protokoll. Az SMB-t úgy tervezték, hogy különféle protokollrendszerek – többek között az IPX/SPX (ez a NetWare régi protokollverme), a NetBEUI (ez a helyi PC-hálózatok elavult protokollja) és a TCP/IP – felett legyen képes működni.

Más hálózati protokollokhoz hasonlóan az SMB is egy ügyfélre (egy szolgáltatásokat igénylő számítógépre) és egy kiszolgálóra (egy szolgáltatásokat nyújtó számítógépre) épül. Minden munkamenet előzetes információcserével indul, amelynek során a felek megállapodnak az SMB-nyelvjárásban, a kiszolgáló pedig azonosítja és belépteti az ügyfelet. Az azonosítási (hitelesítési) eljárás részletei operációs rendszerenként, illetve a beállításoktól függően különbözhetnek, de az SMB oldaláról a bejelentkezést mindig egy `sesssetupX` SMB zárja magába. (Az SMB protokoll hatálya alatti átvitelt egyszerűen SMB-nek hívják.)

Ha a bejelentkezés sikeres, az ügyfél egy olyan SMB-t küld, amelyben megadja annak a megosztott hálózati erőforrásnak a nevét, amelyet el szeretne érni. Ha a hozzáférés sikeres, az ügyfél megnyithatja, bezárhatja, olvashatja és írhatja a hálózati erőforrást, a kiszolgáló pedig elküldi a kérelem teljesítéséhez szükséges adatokat.

Az SMB-t általában Windows-protokollnak tekintik, és az SMB jelentősége nagyrészt valóban abban áll, hogy szorosan beépül a Windows-ügyfelek felhasználói felületébe. Az SMB-nek ugyanakkor létezik egy nyílt szabványú változata, a CIFS (Common Internet File System, közös internetes fájlrendszer). Az SMB és a CIFS protokollok részleteit a fejlesztők jól ismerik, és más operációs rendszerek is támogatják azokat a kiszolgálókat, amelyek az SMB segítségével kommunikálnak a Windows-ügyfelekkel. Az egyik népszerű nyílt forrású kiszolgálóprogram, a Samba (a név hasonlósága az SMB-hez nem véletlen), például Unix/Linux rendszerek számára kínál SMB-fájlszolgáltatásokat.

Összefoglalás

A TCP/IP-kapcsolati segédprogramok eszközkészlete segít a felhasználóknak a hálózati kapcsolatok beállításában és hibáinak elhárításában. Mindegyik segédprogram csak kis mennyiségű információt jelenít meg, de ha a felhasználó tudja, hogyan használja ezeket az eszközöket, gyorsan rábukkanhat a problémák forrására, és időben elháríthatja a fejfájást. Ebben az órában ezek közül a TCP/IP-segédprogramok közül tekintettünk át néhányat, valamint azokkal a segédprogramokkal is megismerkedtünk, amelyek fájlok átvitelére, illetve távoli könyvtárak tallózására szolgálnak.

Kérdezz-felelek

K Melyik segédprogram jeleníti meg az adatcsomagok útját?
V A `traceroute`, amely a Windows rendszereken `tracert` néven ismeretes.
K Melyik segédprogram jelenít meg statisztikákat a TCP/IP-protokollokról?
V A `netstat`.
K Melyik segédprogram teszi lehetővé egy adott IP-címmel létesített kapcsolat ellenőrzését?
V A `ping`.
K Mi az FTP alapértelmezett ábrázolása (átviteli típusa)?
V ASCII.
K Általában mely FTP-parancsok nem megengedettek, amikor egy felhasználó névtelen fiókon keresztül kapcsolódik?
F Az `anonymous` (névtelen) fiókot általában úgy állítják be, hogy csak olvasási hozzáférést engedélyez, tehát a fájlba író vagy az FTP-kiszolgálón a könyvtárszerkezetet módosító parancsok nem engedélyezettek. Ilyen parancs például a `put`, az `mkdir`, az `rmdir`, az `mput` és az `mget`.

K Ki lehet íratni egy könyvtár fájljainak listáját a TFTP segítségével?
F Nem. A TFTP csak átvinni képes a fájlokat, távoli könyvtárat nem tekinthetünk meg vele.
K Milyen előnyei vannak az RCP-nek az FTP-vel szemben?
F Egyszerűbb a nyelvtana, valamint nincs szükség bejelentkezésre a fájlok másolásához. A felhasználó alapú hitelesítésnek ez a hiánya ugyanakkor az RCP egyik legfőbb hátránya is.

Gyakorlat

Hajtsuk végre az alábbi parancsokat, és tekintsük meg a kapott válaszokat a számítógépünkön:

`ipconfig /all` vagy `ifconfig -a` (Nem minden TCP/IP-verem valósítja meg ezeket.)
`ping 127.0.0.1`
`ping` *w.x.y.z* – A *w.x.y.z* helyére írjuk a számítógépünk IP-címét.
`ping` *w.x.y.z* – A *w.x.y.z* helyére írjuk egy másik helyi számítógép IP-címét.
`ping` *w.x.y.z* – A *w.x.y.z* helyére írjuk az alapértelmezett átjárónk IP-címét.
`ping` *w.x.y.z* – A *w.x.y.z* helyére írjuk egy távoli számítógép IP-címét.
`ping localhost`
`ping http://www.whitehouse.gov` (Ehhez kapcsolódnunk kell az Internetre, és rendelkeznünk kell egy DNS-kiszolgálóval.)
`hostname`
`ping` *<állomásnév>* – Az *<állomásnév>* helyére írjuk a számítógépünk tényleges állomásnevét.
`arp -a` vagy `arp -g` – Legalább az egyik vagy mindkettő is működhet. Várjunk pár percet, és ismételten adjuk ki a parancsot.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **`arp`** – Segédprogram, amely a címfeloldási táblázat (ARP-, vagyis Address Resolution Protocol tábla) tartalmának beállítására és megjelenítésére szolgál.
- **biztonságos másolás (`scp`)** – Az `scp` az `rcp` biztonságos változata, amely az SSH-n keresztül adattitkosítást biztosít.
- **FTP (File Transfer Protocol, fájlátviteli protokoll)** – Két számítógép között fájlok átvitelére szolgáló protokoll, illetve ügyfél- és kiszolgálóprogram. A fájlok átvitelén kívül az FTP segédprogrammal könyvtárakat hozhatunk létre és törölhetünk, valamint megjeleníthetjük azok tartalmát.

- hálózati csomaglehallgató – Diagnosztikai alkalmazás vagy hardvereszköz, amely képes elfogni és megjeleníteni az adatcsomagok tartalmát.
- `hostname` – Segédprogram, amely a helyi állomás állomásnevét jeleníti meg.
- `ifconfig` – Unix/Linux-segédprogram, amely a TCP/IP-beállítások megjelenítésére szolgál.
- `ipconfig` – Windows-segédprogram, amely a TCP/IP-beállítások megjelenítésére szolgál.
- `nbtstat` – Segédprogram, amely statisztikákat és más diagnosztikai információkat nyújt a TCP/IP feletti NetBIOS-ról.
- `netstat` – Segédprogram, amely statisztikákat és más diagnosztikai információkat nyújt a TCP/IP-protokollokról.
- NFS (Network File System, hálózati fájlrendszer) – Az NFS lehetővé teszi egy NFS-ügyfélgép felhasználójának, hogy úgy férjen hozzá egy távoli NFS-kiszolgáló fájljaihoz, mintha azok helyi fájlok lennének.
- `ping` – Diagnosztikai segédprogram, amellyel egy másik állomással létesített kapcsolat működését vizsgálhatjuk.
- `route` – Segédprogram, amely az útválasztási táblázatok tartalmának beállítására és megjelenítésére szolgál.
- SFTP (Secure File Transfer Protocol, biztonságos fájlátviteli protokoll) – Az FTP biztonságos változata, amely az SSH-n keresztül adattitkosítást biztosít.
- SMB (Server Message Block, kiszolgálói üzenetblokk) – Az SMB egy alkalmazásrétegbeli protokoll, amely lehetővé teszi a Windows rendszerű ügyfeleknek, hogy olyan hálózati erőforrásokhoz férjenek hozzá, mint a fájlok és a nyomtatók.
- távmásolás (`rcp`) – Az `rcp` egy Unix alapú segédprogram, amely fájlok másolását teszi lehetővé számítógépek között, a Unix `cp` parancsához hasonló utasításforma használatával. Az `rcp` egyszerű módot ad a fájlok másolására, és nem igényel bejelentkezést a felhasználó részéről a fájlmásolási művelet kezdeményezése előtt.
- TFTP (Trivial File Transfer Protocol, egyszerű fájlátviteli protokoll) – UDP alapú protokoll, illetve ügyfél- és kiszolgálóprogram, amely egyszerű fájlátviteli műveletek végrehajtására használatos.
- `traceroute` – Segédprogram, amely megjeleníti egy csomag útválasztási útvonalát a forrástól a célig.
- `tracert` – A `traceroute` segédprogram Microsoft-megfelelője.

15. ÓRA

Hálózatfigyelés és távoli hozzáférés

A fejezet tartalmából:

- Telnet
- A Berkeley r* segédprogramok
- Megbízható hozzáférés
- Hálózatkezelés
- Az SNMP
- Az RMON

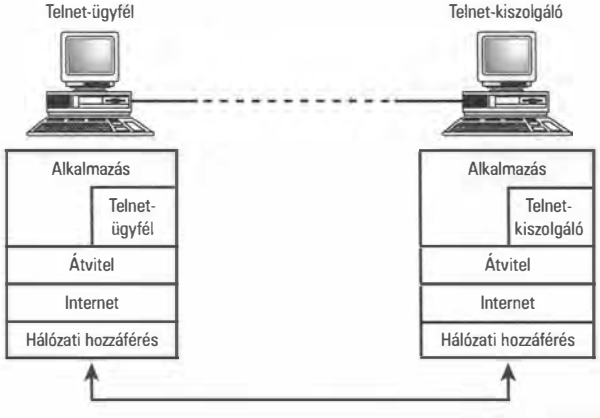
A hálózatok az erőforrások távoli megosztására valók, ezért szinte minden, amit egy hálózaton csinálunk, belefér a távoli hozzáférés meghatározásába. Mindazonáltal csak néhány TCP/IP-segédprogramot tekintenek hagyományosan távoli hozzáférési eszköznek. Ezek a segédprogramok a Unix világában születtek, de azóta más operációs rendszerekre is átültették őket, és arra szolgálnak, hogy a távoli felhasználóknak a helyi felhasználókéhoz hasonló lehetőségeket biztosítsanak. Ezen az órán olyan eszközökkel ismerkedünk meg, mint a Telnet, a Berkeley r* segédprogramok, valamint az SSH.

Az óra végeztével a következőkre leszünk képesek:

- El tudjuk magyarázni a Telnet célját.
- Fel tudunk sorolni néhányat a Berkeley r* segédprogramjai közül.
- El tudjuk magyarázni, hogyan működik a megbízható hozzáférés biztonsági rendszere.

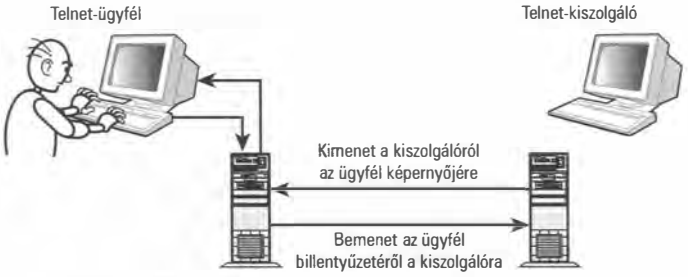
Telnet

A Telnet olyan összetevők gyűjteménye, amelyek terminálszerű hozzáférést adnak egy távoli számítógéphez. A Telnet-munkamenetekhez egy Telnet-ügyfél szükséges, amely távoi terminálként szolgál, valamint egy Telnet-kiszolgáló, amely fogadja a kapcsolati kérelmet, és engedélyezi a kapcsolódást. Ezt a viszonyt a 15.1. ábra szemlélteti.



15.1. ábra
Telnet-kiszolgáló és -ügyfél

A Telnet egyben protokoll is, tehát egy szabályrendszer, amely meghatározza a Telnet-kiszolgálók és -ügyfelek közötti műveleteket. A Telnet protokollt több RFC-dokumentum írja le. Mivel a Telnet egy jól meghatározott, nyílt protokollon alapul, hardver- és szoftverrendszerek széles körén valósítható meg. A Telnet alapvető célja az, hogy lehetőséget adjon arra, hogy egy távoli felhasználó billentyűparancsok begépelésével a hálózaton keresztül bemenetet biztosíthasson egy másik számítógépnek. A munkamenethez tartozó képernyőkimenet a távoli számítógépről (a kiszolgálóról) aztán a hálózaton keresztül az ügyfélrendszerre kerül (lásd a 15.2. ábrát). A távoli felhasználó tehát lényegében úgy adhat ki parancsokat a kiszolgálónak, mintha helyben lenne bejelentkezve.



15.2. ábra

Hálózati bevitel és kivitel a Telnet segítségével

Unix rendszereken a `telnet` parancsot a parancssorba kell beírni, a következő alakban:

`telnet` *`állomásnév`*

Az *`állomásnév`* annak a számítógépnek a neve, amelyhez kapcsolódni szeretnénk. (Állomásnév helyett IP-címet is beírhatunk.) A fenti parancs elindítja a Telnet alkalmazást, és amikor az alkalmazás fut, az általunk beírt parancsok a távoli számítógépen hajtódnak végre. A Telnet emellett különleges parancsokat is biztosít, amelyeket a Telnet-munkamenetek során használhatunk:

- `close` – Ezzel a paranccsal zárhatjuk be a kapcsolatot.
- `display` – Ezzel a paranccsal a kapcsolat beállításait jeleníthetjük meg, például a használt kaput vagy terminálutánzást.
- `environ` – Ezzel a paranccsal a környezeti változókat állíthatjuk be. A környezeti változókat az operációs rendszer használja a gépre vagy a felhasználóra jellemző információk megadására.
- `logout` – Ez a parancs kijelentkezteti a távoli felhasználót, és bezárja a kapcsolatot.
- `mode` – Ezzel a paranccsal válthatunk az ASCII és a bináris fájlátviteli mód között. Az ASCII módot a szövegfájlok hatékony átvitelére tervezték, míg a bináris módot az egyéb típusú állományokhoz, például a végrehajtható programfájlokhoz és a képfájlokhoz.
- `open` – Ezzel a paranccsal kapcsolódhatunk egy távoli számítógéphez.
- `quit` – Ezzel a paranccsal léphetünk ki a Telnet alkalmazásból.
- `send` – Ezzel a paranccsal különleges Telnet-protokollutasításokat küldhetünk a távoli számítógépnek, például megszakítási parancsot, sortörést vagy fájlvége-jelzést.
- `set` – Ezt a parancsot a kapcsolati beállítások megadására használhatjuk.
- `unset` – Ezzel a paranccsal törölhetjük a kapcsolati paramétereket.
- `?` – Ezzel a paranccsal segítséget kérhetünk.

Az olyan grafikus rendszereken, mint a Microsoft Windows, a Telnet alkalmazásnak saját ikonja lehet, és ablakban futhat, a háttérben megbúvó parancsok és folyamatok azonban ugyanazok, mint a szöveges rendszereken. Ha erről többet szeretnénk tudni, olvassuk el a gyártó leírását.

A Telnet valaha rendkívül hasznos eszköz volt, az utóbbi években azonban felváltották az olyan biztonságosabb lehetőségek, mint az SSH, amelyről az óra későbbi részében tanulunk. A Telnettel az az egyik gond, hogy pontosan azt adja meg a hálózati támadóknak, amire a legjobban vágynak: közvetlen hozzáférést egy terminál-munkamenethez egy távoli kiszolgálón. Ráadásul annak ellenére, hogy a Telnet-szabvány támogatja a jelszavas hitelesítést, a jelszavak átvitele általában sima szövegként történik. Mindazonáltal a Telnet ismerete fontos a hálózatok fejlődésének megértése szempontjából, így a TCP/IP tárgyalása sem lehet teljes anélkül, hogy szót ne ejtenénk róla.

Berkeley r* segédprogramok

A Berkeley Systems Design (BSD) Unix-megvalósítás – amelyet röviden csak BSD Unixként ismernek – fontos lépés volt a Unix fejlődésében. Számos újítás, amely először a BSD Unixban jelent meg, ma már szabványos része a többi Unix rendszernek, és más operációs rendszerekbe is beépítették a TCP/IP és az Internet világában.

A BSD Unix újításainak egyikét az a néhány parancssori segédprogram jelentette, amelyet a távoli hozzáféréshez fejlesztettek ki. A segédprogramoknak ez a kis csoportja „Berkeley r* segédprogramok" néven vált ismertté, mivel mindegyik segédprogram neve *r*-rel kezdődik (*r*, mint *remote*, vagyis *távoli*). A Berkeley r* segédprogramok változatai ma is elérhetők a Unix, Linux és Windows rendszereken, annak ellenére, hogy a Telnethez hasonlóan biztonsági szempontból ma már ezek az eszközök is elavultnak számítanak.

A legfontosabb Berkeley r* segédprogramok a következők:

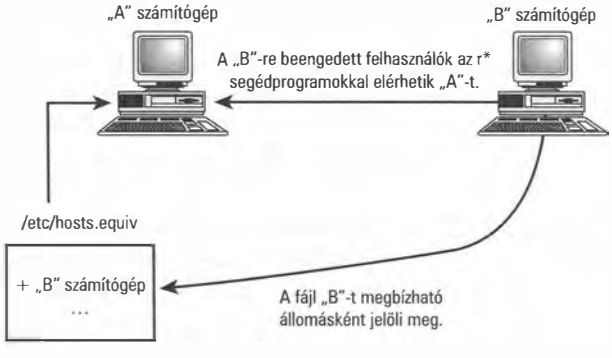
- **`rlogin`** – Ez a program teszi lehetővé a felhasználók távoli bejelentkezését.
- **`rcp`** – Ez a program a távoli fájlátvitelt teszi lehetővé.
- **`rsh`** – Ez a program egy távoli parancsot hajt végre az `rshd` démonon keresztül.
- **`rexec`** – Ez a program a távoli parancsokat az `rexecd` démonon keresztül hajtja végre.
- **`ruptime`** – Ezzel a paranccsal a rendszer üzemidejéről és a kapcsolódott felhasználók számáról jeleníthetünk meg rendszerinformációkat.
- **`rwho`** – Ezzel a paranccsal a jelenleg kapcsolódott felhasználókról jeleníthetünk meg információkat.

Az r* segédprogramokat a TCP/IP-hálózatok történetének korábbi, egyszerűbb szakaszában tervezték, ezért az alkotóik arra számítottak, hogy csak megbízható felhasználók fognak hozzájuk férni. Ma már a legtöbb rendszergazda egyáltalán nem is ismer olyat, hogy „megbízható felhasználó". Az r* segédprogramok használatát általában túl kockázatosnak tartják a mai nyílt, összekapcsolt hálózatokon, és még egy belső hálózaton is körültekintőnek kell lennünk, hogy mikor és hogyan használjuk őket. Az r* segédprogramok ugyanakkor rendelkeznek egy kezdetleges biztonsági rendszerrel, amely megfelelő megvalósítás esetén bizonyos fokú védelmet nyújt egy korlátozott és megbízhatónak minősített környezetben.

Az r* segédprogramok a *megbízható hozzáférés* elvére támaszkodnak. A megbízható hozzáférés lényegében azt jelenti, hogy az egyik számítógép megbízik egy másik számítógép hitelesítési rendszerében. A 15.3. ábrán az „A" számítógép *megbízható állomásként* jelöli meg a „B" számítógépet, így a „B" számítógépre bejelentkező felhasználók az r* segédprogramok használatával anélkül férhetnek hozzá az „A" számítógéphez, hogy jelszót kellene megadniuk. Az „A" számítógép egyes felhasználókat is megjelölhet *megbízható felhasználóként*. A megbízható állomásokat és felhasználókat az `/etc/hosts.equiv` fájl azonosítja azon a távoli gépen, amelyhez hozzáférést szeretnének kapni, de a felhasználók kezdőkönyvtárában található `.rhosts` fájlban is megadható a megbízható hozzáférés az adott felhasználó fiókja számára.

Mivel az `/etc/hosts.equiv` és az `.rhosts` fájl rendszer-erőforrásokhoz nyújt hozzáférést, kiemelt célpontot jelentenek a hálózati támadók számára. Ezeknek a fájloknak a sebezhetősége az egyik oka annak, amiért az r* segédprogramokat ma már nem tartják biztonságosnak.

A következőkben részletesebben is bemutatunk néhányat a Berkeley r* segédprogramok közül.



15.3. ábra

Unix típusú megbízható hozzáférés

rlogin

Az `rlogin` egy távoli bejelentkezési segédprogram. Az `rlogin` segítségével egy olyan Unix-állomáshoz kapcsolódhatunk, amelyen fut az `rlogind` kiszolgálódémon (a `d` a daemon, vagyis démon rövidítése). Az `rlogin` ugyanazt a célt szolgálja, mint a Telnet, de lényegesen kevésbé sokoldalú. Az `rlogin`-t kifejezetten Unix rendszerek elérésére tervezték, míg a Telnetnek, amelyet egy TCP/IP-szabvány ír le, szélesebb körű alkalmazása lehetséges. Ezenkívül az `rlogin` a Telnetben elérhető egyes beállításegyeztetési lehetőségekkel sem rendelkezik. Az `rlogin` egyik lényeges szolgáltatása, hogy mivel az r* segédprogramok biztonsági modelljét alkalmazza, megengedi a jelszó nélküli távoli bejelentkezést. A jelszó nélküli hozzáférést minden r* segédprogram biztosítja, de vannak felhasználók, akik a jelszó nélküli terminál-munkameneteket valamivel nyugtalanítóbbnak tartják, mint más szolgáltatásokat, amelyek elérhetők az r* segédprogramokon keresztül. Mindazonáltal az r* segédprogramok biztonsági modellje a hozzáférést nem korlátozza a megbízható felhasználókra.

Fontos észben tartanunk, hogy sok hálózati operációs rendszer lehetővé teszi valamilyen módon a hálózati erőforrások jelszó nélküli elérését, miután a felhasználó átesett valamilyen kezdeti azonosításon. A 23. órán terítékre kerülő Kerberos hitelesítési rendszer például Unix/Linux- és Windows-hálózatokon is biztosítja a hálózati erőforrások jelszó nélküli elérését. Az r* segédprogramok számos előnyét tehát ma már más, biztonságosabb módszerek segítségével is kihasználhatjuk.

Az `rlogin` utasításformája a következő:

```
rlogin állomásnév
```

Az *állomásnév* annak a számítógépnek az állomásneve, amelyhez hozzáférést szeretnénk szerezni. Ha nem adunk meg felhasználónevet, a felhasználónév alapértelmezés szerint a helyi számítógépre bejelentkezett felhasználó neve lesz. Más felhasználónevet az alábbi módon adhatunk meg:

```
rlogin állomásnév -l felhasználónév
```

Itt a *felhasználónév* az a felhasználónév, amelyet a bejelentkezéshez használni szeretnénk.

Az `rlogind` kiszolgálódémon, amelynek futnia kell a kiszolgálógépen, ez után belenéz a `hosts.equiv` és `.rhosts` fájlokba, hogy ellenőrizze az állomás és a felhasználó adatait. Ha ez a hitelesítés sikeres, a távoli munkamenet megkezdődhet.

rcp

Az `rcp` segédprogram távoli fájlhozzáférést biztosít. Az `rcp` nem olyan sokoldalú vagy széles körben használatos, mint az FTP, de azért néha használják fájlok átvitelére. Az `rcp`-ről a 14. fejezetben bővebben is szót ejtettünk.

rsh

Az `rsh` segédprogram segítségével egyetlen parancsot hajthatunk végre egy távoli számítógépen, anélkül, hogy be kellene jelentkeznünk oda. Az `rsh` a *remote shell* (távoli héj) rövidítése. (A *héj* egy parancsfelület, amelyen keresztül parancsokat adhatunk az operációs rendszernek.) Az `rshd` démon, amely a távoli számítógépen fut, fogadja az `rsh`-parancsot, ellenőrzi a felhasználó és az állomás nevét, és végrehajtja a parancsot. Az `rsh`-nak akkor vehetjük hasznát, ha egyetlen parancsot szeretnénk kiadni, és nem kívánunk terminál-munkamenetet létesíteni a távoli számítógéppel.

Az `rsh` parancs utasításformája a következő:

```
rsh -l felhasználónév állomásnév parancs
```

Itt az *állomásnév* a távoli számítógép állomásneve, a *felhasználónév* a távoli számítógép eléréséhez használandó felhasználó neve, a *parancs* pedig a végrehajtani kívánt parancs. A felhasználónév (amelyet az `-l` kapcsoló előz meg), elhagyható. Ha nem adunk meg felhasználónevet (lásd alább), az alapértelmezés a helyi gépen használt felhasználónév lesz:

```
rsh állomásnév parancs
```

rexec

Az `rexec` annyiban az `rsh`-hoz hasonlít, hogy egy parancs végrehajtására utasítja a távoli számítógépet. Az `rexec` az `rexecd` démonra támaszkodik. A parancs utasításformája a következő:

```
rexec állomásnév -l felhasználónév parancs
```

Az *állomásnév* itt is a távoli számítógép állomásneve, a *felhasználónév* a távoli számítógépen használandó felhasználói fiók neve, a *parancs* pedig a végrehajtani kívánt parancs. Ha az `-l` *felhasználónév* részt elhagyjuk, a felhasználónév alapértelmezés szerint a helyi gépen használt felhasználónév lesz.

ruptime

Az `ruptime` a hálózat egyes számítógépeire bejelentkezett felhasználók számát összegzi, valamint azt is kiírja, hogy az egyes számítógépek mennyi ideje üzemelnek (a parancs neve erre utal), illetve egyéb rendszerinformációkat is megjelenít.

Ha jelentést szeretnénk készíttetni az `ruptime`-mal, csak ennyit kell beírnunk:

```
ruptime
```

Mind az `ruptime`, mind az `rwho` (lásd alább) az `rwhod` démonra támaszkodik. Valójában a hálózat minden számítógépe rendelkezik egy `rwhod` démonnal, amely rendszeres jelentést sugároz a felhasználók tevékenységéről, és minden `rwhod` démon megkapja és elraktározza a többi `rwhod` démon jelentéseit, hogy a felhasználói tevékenységről az egész hálózatra kiterjedően nézetet kapjunk.

rwho

Az `rwho` a jelenleg a hálózat számítógépeire bejelentkezett összes felhasználóról szolgáltat adatokat. A program felsorolja a felhasználóneveket, azt, hogy melyik felhasználó melyik számítógépre jelentkezett be, a bejelentkezés idejét, valamint a bejelentkezés óta eltelt időt.

Az `rwho` parancs utasításformája egyszerűen az alábbi:

```
rwho
```

Az alapértelmezett jelentésből kimaradnak azok a felhasználók, akiknek a terminálja több mint egy órája inaktív. Ha minden felhasználóról jelentést szeretnénk kapni, használjuk az `-a` kapcsolót:

```
rwho -a
```

Az `rwho` az `ruptime`-hoz hasonlóan az `rwhod` démont használja.

SSH

Ahogy már valószínűleg tudjuk ebből a fejezetből, a távoli hozzáférés olyan klasszikus TCP/IP-segédprogramjai, mint a Telnet vagy az r* eszközök, nem megfelelőek az olyan környezetekben, ahol a biztonság lényeges. Az r* segédprogramokat ma már egyre kevésbé használják, a Telnet használata azonban néhány területen – így a betárcsázós hozzáférésnél vagy a védett hálózatok távoli felügyeleténél – megmaradt, még ha a legtöbb informatikus szakember véletlenül sem nyúlna a Telnethez a nyílt Interneten.

Az Internet fejlődése ugyanakkor még inkább középpontba állította a hálózatokat és a távoli hozzáférést. A mai hálózatokon a távoli héj-munkameneteket jellemzően azoknak a protokolloknak és segédprogramoknak a segítségével kezelik, amelyeket összefoglaló néven Secure Shellnek (SSH, biztonságos héj) hívnak. Az SSH lényegében egyenértékű a Berkeley r* segédprogramok megvalósításával, csak éppen nyilvános kulcsú titkosítást és a Secure Sockets Layer (SSL) használatával biztonságos hálózatkezelést nyújt. Az SSL-ről és a nyilvános kulcsú titkosításról a 23. órán tanulunk majd részletesebben.

Az SSH-csomag főbb összetevői a következők:

- SSH – Távoli héjprogram, amely az `rlogin`, az `rsh` és a `telnet` felváltására hivatott.
- `scp` – Fájlátviteli segédprogram, amelyet az `rcp` felváltására terveztek.
- `sftp` – Fájlátviteli segédprogram, amely az FTP helyett használható.

Az SSH legnépszerűbb megvalósítása a nyílt OpenSSH projekt. Az OpenSSH Unix, Linux, Windows, Mac OS, sőt Palm OS rendszerre is elérhető, és számos további segédprogram is rendelkezésre áll hozzá, amelyekkel a kulcsaláírásokat és a titkosítást kezelhetjük. Az SSH-kapcsolatok kiszolgálóoldalán az `sshd` (SSH-démon) áll, amelyet az OpenSSH-csomag szintén tartalmaz.

Az OpenSSH használatával a következőképpen jelentkezhetünk be egy távoli rendszerre:

```
ssh felhasználó@állomásnév
```

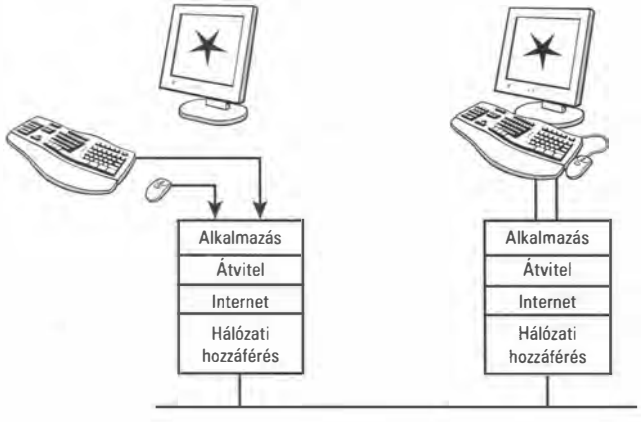
A készenléti jelnél ez után meg kell adnunk a jelszavunkat; ezt követően ugyanúgy dolgozhatunk, mint egy helyi parancshéjban. Az SSH sokkal biztonságosabb az Interneten, mint az elődei, mert a beépített titkosítás a kémkedés és a hamisítás legtöbb formája ellen védelmet nyújt. Sok tűzfalprogram lehetővé teszi, hogy a belső hálózatot SSH-kapcsolaton keresztül érjék el kívülről, így a hálózati rendszergazda az SSH segítségével az Interneten át is bejelentkezhet a belső hálózatra.

A biztonságos távoli héjkapcsolatok mellett az SSH a kapuátirányítás egy formáját is támogatja, hogy a nem biztonságos alkalmazások is biztonságosan működhessenek az SSH alapú titkosított kapcsolatokon keresztül.

Képernyőmegosztás

Sok rendszergazda és haladó felhasználó jobban szeret a parancshéjban dolgozni, ahol egyetlen sornyi szövegnek egyetlen válasz felel meg. A parancshéj ezenkívül az olyan eszközök segítségével, mint a Telnet, az `rsh` vagy az SSH, könnyen távoli végrehajtási környezetté bővíthető. Mindazonáltal a legtöbb felhasználó ma már nem a parancshéjból dolgozik, hanem egérrel kattintgat egy grafikus felhasználói felületen.

Grafikus felhasználói felületen keresztül távoli hozzáférést nyújtani kissé bonyolultabb, de az elv ugyanaz (lásd a 15.4. ábrát). Az „A" számítógép alkalmazásrétegében működő szoftverösszetevő elfogja a billentyűzetről érkező bemenetet, és átirányítja a protokollvermen keresztül a „B" számítógépre, a „B" számítógép pedig ezt követően visszaküldi a képernyőre írandó adatokat a hálózaton át az „A" számítógépnek. Végeredményben tehát az „A" számítógép billentyűzete és egere a „B" számítógép billentyűzeteként és egereként működik, az „A" számítógép képernyője pedig a „B" számítógép Asztalának nézetét mutatja. Röviden, az „A" számítógépnél ülő felhasználó távvezérelheti a „B" számítógépet.



15.4. ábra

A grafikus felhasználói felületű távoli hozzáférési eszközök átirányítják a billentyű- és egérparancsokat

A grafikus felületű távoli hozzáférést eredetileg olyan külső eszközök tették népszerűvé, mint a Symantec pcAnywhere vagy a Netopia Timbuktu programja. A Mac OS és a Windows legújabb változatai közvetlenül az operációs rendszerbe beépített távoli hozzáférési lehetőségekkel rendelkeznek – ilyen eszköz az Apple Remote Desktop és a Windows Vista Remote Desktop Connection (Távoli asztal) programja. A Unix/Linux rendszerek mindig is rendelkeztek ennek a szolgáltatásnak egy kezdetleges változatával az X Server grafikus környezet alapszerkezetében, az olyan újabb eszközök azonban, mint a VNC (Virtual Network Computing) vagy a NoMachine NX programja, kényelmesebbé tették a távoli hozzáférést a végfelhasználó számára.

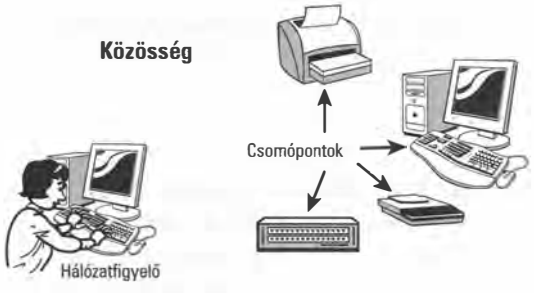
A távolról dolgozó rendszergazdák és informatikai ügyfélszolgálatok gyakran használnak képernyőmegosztó eszközöket az asztali PC-k beállítására és hibaelhárítására.

SNMP

A protokollok feladata a kommunikáció lehetővé tétele, és amikor egy adott *típusú* kommunikációnak jellegzetes és meghatározható tulajdonságai vannak, valószínűleg találunk hozzá való protokollt. Az SNMP-t (Simple Network Management Protocol, egyszerű hálózatkezelési protokoll) hálózatokon található távoli eszközök kezelésére és figyelésére tervezték. Az SNMP lehetővé teszi, hogy egy rendszerben egyetlen hálózati rendszergazda, egyetlen munkaállomásról, távolról kezelje és felügyelje a számítógépeket, útválasztókat és más hálózati eszközöket.

Az SNMP rendszerének főbb összetevői, amelyeket a 15.5. ábrán láthatunk, a következők:

- *Hálózatfigyelő* (network monitor) – Felügyeleti konzol, más néven kezelőprogram vagy NMS (Network Management Console, hálózati felügyeleti konzol), amely a hálózaton található eszközök kezelésére szolgáló központi hely. A hálózatfigyelő általában egy átlagos számítógép, amely rendelkezik a szükséges SNMP felügyeleti szoftverrel.
- *Csomópontok* (node) – A hálózaton található eszközök.
- *Közösség* (community) – Csomópontok csoportja egy közös kezelésű környezetben.

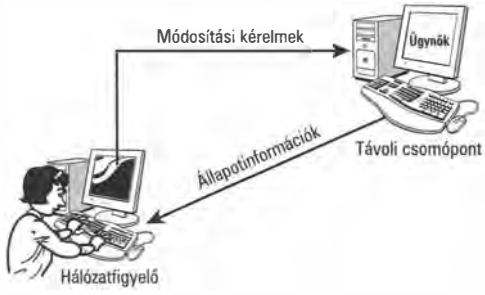


15.5. ábra

Az SNMP-közösségek egy vagy több hálózatfigyelőből és csomópontok gyűjteményéből állnak

Ahogy a könyv más részeiből már megtanultuk, a protokollok kommunikációs sémát biztosítanak, de a tényleges műveletek a kommunikáló eszközökön futó alkalmazások között zajlanak. Az SNMP esetében egy *ügynöknek* (agent) nevezett program fut a távoli csomóponton, és ez kommunikál a hálózatfigyelőn futó kezelőszoftverrel (lásd a 15.6. ábrát).

A figyelő és az ügynök az SNMP protokoll segítségével kommunikál, az SNMP pedig a 161-es és 162-es UDP-kapukat használja. Az SNMP régebbi változatai nem kívántak meg semmilyen biztonsági intézkedést a felhasználók beléptetéséhez, csupán a közösség nevét kellett megadni – ezt hívták *közösségi karakterláncnak* (vagyis ismerni kellett a közösségi karakterláncot). Egyes esetekben az ügynököt úgy is be lehetett állítani, hogy csak bizonyos IP-címekről fogadjon adatokat. Ez a fajta biztonsági rendszer azonban a mai követelményeknek már nem felel meg. Az SNMP legújabb változata, az SNMP v3 ezért hitelesítést, bizalmasságot és általánosságban is nagyobb biztonságot nyújt a rendszer számára.



15.6. ábra
A távoli csomóponton futó ügynökprogram információkat küld a csomópontról a hálózatfigyelőnek, és fogadja a beállítások módosítására irányuló kérelmeket

Felmerülhet bennünk a kérdés, hogy *miről* társaloghat a figyelő és az ügynök? Milyen adatok haladnak közöttük az SNMP-n keresztül? Ahogy a következő részből megtudhatjuk, az SNMP rengeteg felügyeleti paramétert határoz meg. A hálózatfigyelő ennek a *felügyeleti információs alapnak* (Management Information Base, MIB) a paramétereit használja arra, hogy információkat kérjen az ügynöktől, és módosítsa a beállításokat.

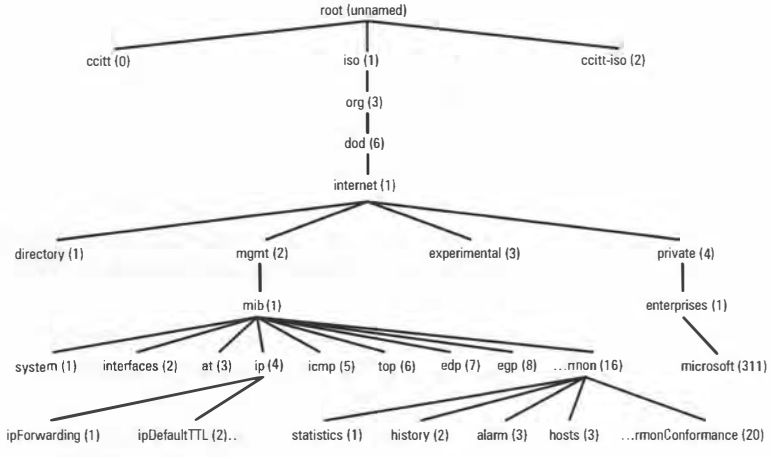
Az SNMP címtér

Az SNMP-kommunikáció alapját az jelenti, hogy a figyelő- és az ügynökszoftver egyaránt képes információt cserélni az MIB bizonyos címezhető területeiről. Az MIB, amelyet a 15.7. ábrán láthatunk, teszi lehetővé a figyelőnek és az ügynöknek, hogy pontos és egyértelmű módon cseréljenek információt. A figyelőnek és az ügynöknek azonos MIB-szerkezetre van szükségük, mert képesnek kell lenniük egyedileg azonosítani egy adott információegységet.

Az MIB hierarchikus címtér, amely minden információegység számára egyedi címet biztosít. Megjegyzendő, hogy a hálózati címekkel ellentétben az MIB-címek nem helyet vagy tényleges eszközt jelölnek; az MIB paraméterek hierarchikusan címtérbe rendezett gyűjteménye. Ez a hierarchikus címelrendezés biztosítja, hogy minden SNMP-eszköz ugyanúgy hivatkozzon egy adott beállításra. Ez a megközelítés a kényelmes decentralizációt is lehetővé teszi; egy adott gyártó például meghatározhatja a saját termékeire érvényes MIB-beállításokat (ezekre gyakran egyszerűen MIB-kként szoktak hivatkozni),

egy szabványügyi szervezet pedig az MIB-fának azt a részét felügyelheti, amelyik a szabványához kapcsolódik. Az MIB minden egyedi címet pontozott jelöléssel ad meg az MIB-objektumon belül.

Az MIB-t több RFC-dokumentum írja le, például az RFC 1158 és az RFC 1213. Az SNMP hivatalos leírását az RFC 1157-ben találhatjuk, míg a legfrissebb SNMP v3-ét az RFC 2570-ben, valamint néhány további RFC-dokumentumban.



15.7. ábra
Egy kis részlet az MIB-ből

Az MIB-n belül címezhető helyek többsége számlálókra hivatkozik, amelyek nyilvánvalóan számértékűek. Ilyen számláló például az `ipForwarding`, amelyet a 15.7. ábrán is megtalálhatunk, illetve az ábrán nem szereplő `ipInReceives`, amely a hálózati szoftver elindítása vagy a számláló utolsó visszaállítása óta beérkezett IP-adatcsomagok számát tartja nyilván.

Az MIB-információk formátuma többféle lehet: számértékű, szöveges, IP-cím, és így tovább. Az MIB beállítási információk egy másik példája az `ipDefaultTTL` beállítás, amely a számítógépekről érkező összes IP-adatcsomagba beszúrt élettartam (TTL, Time To Live) paraméter számértékét tartalmazza.

Az MIB-szerkezet címzése mindig a gyökérnél kezdődik, és innen halad a hierarchiában lefelé, amíg egyedileg nem azonosítottuk a kiolvasni kívánt beállítást. Például ha az `ipDefaultTTL` és az `ipInReceives` MIB-ket szeretné megcímezni, az SNMP-figyelő a következő MIB-címeket küldi az SNMP-ügynöknek:

```
.iso.org.dod.internet.mgmt.mib.ip.ipDefaultTTL
.iso.org.dod.internet.mgmt.mib.ip.ipInReceives
```

Az MIB-fa minden leveléhez tartozik egy számszerű cím is. Egy MIB-re hivatkozhatunk az alfanumerikus karakterláncával, de a számszerű címével is. A hálózatfigyelő valójában a számszerű formát alkalmazza, amikor információkat kér az ügynöktől:

```
.1.3.6.1.2.1.4.2
.1.3.6.1.2.1.4.3
```

Az MIB-címtér közös rendszert ad, amely biztosítja, hogy a figyelő és az ügynök megbízhatóan hivatkozhasson adott paraméterekre. Ezek az MIB-paraméterek aztán parancsok részét képezik, amelyekről a most következő részben ejtünk szót.

SNMP-parancsok

A hálózatfigyelő ügynökszoftver három parancsot ismer: `get`, `getnext` és `set`. Ezek a parancsok a következőket hajtják végre:

- `get` – A `get` parancs az ügynököt egy adott információegység kiolvasására és visszaadására utasítja az MIB-ből.
- `getnext` – A `getnext` parancs az ügynököt a sorban következő információegység kiolvasására és visszaadására utasítja az MIB-ből. Ezt a parancsot például egy értéktáblázat kiolvasására használhatjuk.
- `set` – A `set` parancs az ügynököt egy állítható paraméter beállítására vagy egy olyan objektum alaphelyzetbe állítására utasítja, mint egy hálózati felület vagy egy adott számláló.

Az SNMP-szoftver valójában többféle működésre is alkalmas, a hálózati rendszergazda igényeitől függően. Az alábbiakban áttekintjük az SNMP viselkedésének különböző típusait:

- A hálózatfigyelő ügynök mindig kérelem–válasz módban működik: kérelmeket fogad a figyelőtől, és válaszokat küld neki. Az ügynök vagy egy `get`, vagy egy `getnext` parancsot kap, és egy információt ad vissza egy címezhető területről.
- Bár nem kötelező, az ügynököket gyakran úgy állítják be, hogy üzenetet küldjenek a hálózatfigyelőnek, ha szokatlan esemény történik. Ezek a kéretlen üzenetek a *csapdaüzenetek* vagy *csapdák* (trap); az elküldésükre akkor kerül sor, ha az ügynökszoftver elfog valamilyen szokatlan eseményt.
  Az SNMP-ügynökszoftver például általában úgy működik, hogy a `set` paranccsal meghatározott küszöbértékek túllépését figyeli. Küszöbtúllépés esetén az ügynök elfogja az eseményt, majd összeállít egy üzenetet a hálózatfigyelő számára, amelyben megadja annak a számítógépnek az IP-címét, ahol az elfogott esemény bekövetkezett, valamint hogy melyik küszöbérték túllépésére került sor.
- Az ügynökök bizonyos műveletek végrehajtására – például egy útválasztó valamelyik kapujának alaphelyzetbe állítására, vagy az események elfogását vezérlő küszöbértékek beállítására – is kaphatnak utasításokat a figyelőtől. Mint már említettük, az állítható paraméterek beállítására, illetve a számlálók és felületek alaphelyzetbe állítására a `set` parancs szolgál.

A következő példa az SNMP által használt kérelem- és válaszparancsokat szemlélteti. A példában az `snmputil` nevű diagnosztikai segédprogramot használjuk, amely egy figyelő utánzását teszi lehetővé. A segédprogramon keresztül parancsokat adhatunk az ügynöknek, amely esetünkben a `192.59.66.200` IP-című számítógépen található, és a `public` nevű közösség tagja. Figyeljük meg a `.0`-t az első két parancs végén – ezt az utótagot akkor használjuk, amikor olyan egyszerű változókból olvasunk, mint a számlálók:

```
D:\>snmputil get 192.59.66.200 public .1.3.6.1.2.1.4.2.0
Variable = ip.ipDefaultTTL.0
Value    = INTEGER - 128

D:\>snmputil getnext 192.59.66.200 public .1.3.6.1.2.1.4.2.0
Variable = ip.ipInReceives.0
Value    = Counter - 11898
```

Sok SNMP-rendszeren az alapértelmezett közösségi név a `public`. A fenti példában a rendszergazdának ezt valami másra kellett volna módosítania, mert ha az alapértelmezett neveket használjuk, előnyt adunk a támadóknak.

Az SNMP hasznos eszköz a hálózati rendszergazdák számára, de nem tökéletes. Az SNMP legfontosabb hiányosságai a következők:

- *Az alsóbb rétegek nem láthatók.* – Az SNMP az UDP felett, az alkalmazásrétegben található, ezért nem látja, mi történik a protokollverem alsóbb rétegeiben, például a hálózati hozzáférés rétegében.
- *Az SNMP működő protokollvermet igényel.* – Az SNMP figyelő- és ügynökszoftvere csak teljesen működőképes TCP/IP-verem jelenlétében képes kommunikálni egymással. Ha hálózati problémák akadályozzák a verem helyes működését, az SNMP nem tud segíteni a hiba elhárításában.
- *Az SNMP jelentős hálózati forgalmat idézhet elő.* – Az SNMP által használt kérelem–válasz eljárás jókora forgalmat idéz elő a hálózaton. Bár a kéretlen csapdaüzenetek elküldésére csak akkor kerül sor, amikor jelentős események történnek, a hálózatfigyelők folyamatos forgalmat okoznak, ahogy információkat kérnek az ügynököktől.
- *Az SNMP túl sok adatot, de túl kevés információt szolgáltat.* – Mivel az MIB-n belül szó szerint címezhető helyek ezrei találhatók, nagyon sok apró információt nyerhetünk ki belőle. Ezeknek az apró részleteknek az elemzése és az egyes gépeken zajló események hasznos összefoglalása azonban erőteljes felügyeleti konzolt kíván.
- *Az SNMP a számítógépekről képet ad, a hálózatról azonban nem.* – Az SNMP-t arra tervezték, hogy adott eszközökről szolgáltasson információkat, a hálózati szakaszon zajló eseményekről azonban nem ad közvetlen képet.

Remote Monitoring

A Remote Monitoring (RMON, távoli hálózatfigyelés) az MIB címtér bővítése, amelyet távoli LAN-ok figyelésére és karbantartására fejlesztettek ki. Az SNMP-től eltérően, amely egyetlen számítógépről szolgáltat információkat, az RMON közvetlenül a hálózatról rögzít adatokat, ezért a LAN egészéről képes információkat nyújtani.

Az RMON MIB az `.1.3.6.1.2.1.16` címen kezdődik (lásd a 15.7. ábrát), és jelenleg húsz csoportra oszlik, `.1.3.6.1.2.1.16.1`-től `.1.3.6.1.2.1.16.20`-ig. Az RMON-t az IETF fejlesztette ki az SNMP hiányosságainak kiküszöbölésére, illetve hogy a távoli LAN-ok hálózati forgalmáról pontosabb képet kaphassunk.

Az RMON-nak két változata létezik, az RMON 1 és az RMON 2:

- RMON 1 – Az RMON 1 az Ethernet és Token Ring hálózatok figyelésére szolgál. Az RMON 1-en belül minden csoportnak a feladata az alsó két rétegnek, például az OSI hivatkozási modell fizikai és adatkapcsolati rétegének (amelyek a TCP/IP modellben a hálózati hozzáférés rétegének felelnek meg) a figyelése. Az RMON 1-et az RFC 1757 írja le, amely az 1991 novemberében közzétett RFC 1271 frissítése.
- RMON 2 – Az RMON 2 mindazokkal a képességekkel rendelkezik, mint az RMON 1, emellett azonban az OSI hivatkozási modell felső öt rétegének a figyelését is lehetővé teszi (ezek a TCP/IP modell internet-, szállítási és alkalmazásrétegeinek felelnek meg). Az RMON 2 leírását az RFC 2021 és az RFC 2034 tartalmazza, amelyeket 1997-ben tettek közzé.

Mivel az RMON 2 a protokollverem felsőbb rétegeit figyeli, az olyan magasabb szintű protokollokról is képes információt nyújtani, mint az IP, a TCP vagy az NFS. Az RMON célja a hálózati forgalomra vonatkozó adatok rögzítése. A hálózati szakaszokat egy RMON-ügynök (vagy vizsgáló, angolul probe) figyeli, és a forgalmi adatokat egy RMON-konzolra továbbítja. Amennyiben a hálózat több szakaszból áll, minden szakaszt más-más ügynök figyel. Az RMON a különféle típusú információkat statisztikai csoportokba gyűjti. Az RMON 1-ben az alábbi csoportok találhatók:

- *Statistics (Statisztika)* – A Statistics csoport statisztikai adatokat tárol, táblázatok formájában, amelyek a vizsgálóhoz kapcsolódó hálózati szakaszhoz tartoznak. A csoport egyes számlálói a csomagok, a többcímes sugárzások, az ütközések, valamint a túl kicsi vagy túl nagy adatcsomagok számát követik nyomon, és így tovább.
- *History (Előzmények)* – A History csoport időszakosan összegyűjtött és későbbi feldolgozásra szánt statisztikai információkat tárol.

- *Alarm (Riasztás)* – Az Alarm csoport az Event csoporttal (lásd alább) együtt működik. Az Alarm csoport rendszeresen statisztikai mintát vesz a vizsgálóban található változókból, és összehasonlítja azokat a beállított küszöbértékekkel. Amennyiben a küszöbértékek túllépésére kerül sor, egy esemény jön létre, amelyet a hálózatfigyelő értesítésére használhatunk.
- *Hosts (Állomások)* – A Hosts csoport a hálózati szakaszon található állomásokra vonatkozó statisztikákat tárolja. Az adatok összegyűjtése az adatcsomagok fizikai forrás- és célcímének vizsgálatával történik.
- *Host Top n (Felső n állomás)* – A Host Top n csoport jelentéseket állít elő a statisztikák alapján egy adott kategória felső n állomásáról. Előfordulhat például, hogy egy hálózatfigyelő azt szeretné tudni, hogy mely állomások szerepelnek a legtöbb adatcsomagban, vagy hogy mely állomások küldik a legtöbb túlméretezett vagy túl kicsi adatcsomagot.
- *Matrix (Mátrix)* – A Matrix csoport egy táblázatot épít fel, amely a hálózaton megfigyelt összes adatcsomag fizikai forrás- és célcímpárját tartalmazza. Ezek a címpárok két cím közötti társalgásokat határoznak meg.
- *Filter (Szűrő)* – A Filter csoport egy bináris minta létrehozását teszi lehetővé, amelyet adatcsomagok keresésére vagy szűrésére használhatunk a hálózaton.
- *Capture (Rögzítés)* – A Capture csoport lehetővé teszi, hogy a Filter csoport segítségével kiválasztott adatcsomagokat a hálózatfigyelő későbbi kiolvasásra és vizsgálatra rögzítse.
- *Event (Esemény)* – Az Event csoport az Alarm csoporttal együttműködve eseményeket hoz létre, amelyek értesítik a hálózatfigyelőt, ha egy megfigyelt objektum valamelyik küszöbértékének túllépésére kerül sor.
- *Token Ring (Vezérjeles gyűrű)* – A Token Ring csoport a vezérjeles gyűrűkre vonatkozó információkat gyűjti össze.

Az RMON 2-ben további csoportok is találhatók, amelyek a felsőbb szintű protokollok áttekintését segítik.

Összefoglalás

Ezen az órán a TCP/IP-hez kifejlesztett távoli hozzáférési segédprogramok közül mutattunk be néhányat. Megismertük a Telnetet, az r* segédprogramokat és az SSH-t. Ezeket a segédprogramokat parancsok végrehajtására és információszerzésre használhatjuk egy távoli számítógépen.

Azt is megtanultuk, hogy az SNMP protokoll fontos szerepet játszik a központosított hálózatfigyelésben és a távoli hálózatok karbantartásában. Egy hálózatfelügyeleti konzolra és egy központi helyre támaszkodva a hálózatfigyelő szoftver értesül róla, ha szokatlan események történnek, és láthatja a hálózati forgalom állapotát, amelyről az útválasztókon, elosztókon és kiszolgálókon működő ügynökök tesznek jelentést.

A hálózatfelügyeleti konzol ezenkívül a hálózatfigyelő számára olyan feladatok végrehajtását is lehetővé teszi, mint a kapuk alaphelyzetbe állítása az útválasztókon, de akár távoli eszközök alaphelyzetbe állítására is lehetőséget ad, ha a kevésbé drasztikus megoldások nem oldanák meg a problémát.

Sok újabb hálózati eszköz tartalmaz beágyazott RMON-szolgáltatásokat. Az RMON jelentősen csökkenti az SNMP esetében szokásos hálózati forgalmat, és nincs szüksége erőteljes hálózatfelügyeleti konzolra az adatok értelmezéséhez. Ugyanakkor az RMON használata jelentős feldolgozási tevékenységet von maga után az RMON-ügynök vagy vizsgáló részéről, amelynek a feladata a hálózati forgalom rögzítése.

Kérdezz-felelek

K *A Telnet kiszolgálóprogram, ügyfélprogram vagy protokoll?*

V A Telnet név a kiszolgálóra, az ügyfélprogramra és a Telnet protokollra is vonatkozhat.

K *Melyik fájlt kell használnunk, ha egy állomást megbízható állomásként szeretnénk megjelölni?*

V Megbízható állomást az `/etc/hosts.equiv` állományban vagy egy felhasználó kezdőkönyvtárának `rhosts` fájljában határozhatunk meg.

K *Melyik segédprogram árulja el, hogy az Ethelred nevű felhasználó jelenleg be van-e jelentkezve a hálózatra?*

V A jelenleg bejelentkezett felhasználókról az `rwho` segédprogrammal jeleníthetünk meg információkat.

K *Az SNMP protokoll melyik szállítási (átviteli) protokollt és mely kapukat használja?*

V Az SNMP elsősorban a 161-es UDP-kaput használja; a 162-es kapun az SNMP-csapdaüzenetek haladnak át.

K *Hogy hívják azokat az üzeneteket, amelyeket az ügynökök akkor küldenek kéretlenül, ha szokatlan esemény következik be?*

V Csapdaüzenet.

K *A TCP/IP modell melyik rétegét figyeli az RMON 1?*

V A hálózati hozzáférés rétegét.

K *A TCP/IP modell mely rétegeit figyeli az RMON 2?*

V Az RMON 2 a protokollverem valamennyi rétegét figyeli.

K *A hálózatom forgalmi szintjének ciklikus változásairól szeretnék adatokat kapni. Az SNMP-t vagy az RMON-t használjam a hálózat figyelésére?*

V Az SNMP elsősorban hálózati eszközök figyelésére való. Az RMON ezzel szemben az adatokat közvetlenül a hálózati átviteli közegről szerzi, ezért általában jobb választás a hálózati forgalom figyelésére.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **hálózatfelügyeleti konzol** – Hálózatfelügyeleti szoftvert futtató munkaállomás, amely nagy, elosztott hálózatok figyelésére, karbantartására és beállítására szolgál.
- **hálózatfigyelő** – A hálózatfelügyeleti konzolok másik neve.
- **megbízható hozzáférés** – Gyenge biztonsági rendszer, amelyben a rendszergazda megjelöli azokat a távoli állomásokat és felhasználókat, akik hozzáférhetnek a helyi rendszerhez.
- **MIB (Management Information Base, felügyeleti információs alap)** – Hierarchikus címtér, amelyet az SNMP-figyelők és -ügynökök használnak. Az MIB-n belül az egyes paraméterekre pontozott jelöléssel hivatkozhatunk, az MIB szerkezet tetejétől lefelé haladva a kívánt MIB-címig.
- **`rcp`** – Távoli fájlátviteli segédprogram.
- **`rexec`** – Távoli parancsvégrehajtási segédprogram.
- **`rlogin`** – Távoli bejelentkezést lehetővé tevő segédprogram.
- **RMON (Remote Monitoring, távoli hálózatfigyelés)** – Az MIB-t bővítő szolgáltatás, amely a hagyományos SNMP-szolgáltatásokhoz képest több lehetőséget nyújt. Ahhoz, hogy az RMON MIB-ben adatokat tárolhasson, az ügynöknek vagy vizsgálónak rendelkeznie kell RMON-szoftverrel.
- **`rsh`** – Távoli parancsvégrehajtási segédprogram.
- **`ruptime`** – Segédprogram, amely az üzemidőről és a kapcsolódott felhasználók számáról jelenít meg rendszerinformációkat.
- **`rwho`** – Segédprogram, amely a jelenleg kapcsolódott felhasználókról jelenít meg rendszerinformációkat.
- **SNMP (Simple Network Management Protocol, egyszerű hálózatkezelési protokoll)** – A TCP/IP-hálózatokon található erőforrások kezelésére használatos protokoll.
- **Telnet** – Távoli terminál-segédprogram.
- **ügynök** – Az állomásokon betöltődő SNMP szoftver, amely képes olvasni az MIB-ből, illetve visszaadni a figyelőknek a kívánt eredményeket. Az ügynökök ezen kívül kéretlen üzeneteket is küldhetnek a figyelőknek, ha jelentős szokatlan eseményeket észlelnek.
- **vizsgáló** – Az ügynökprogramok másik neve. A kifejezést elsősorban az RMON-nal kapcsolatban használják.

V. RÉSZ

A TCP/IP és az Internet

16. ÓRA

Az Internet közelebbről

A fejezet tartalmából:

- Az Internet felépítése
- NAP-k és P●P-k
- URI-k

Az egyre csak növekvő Internet a világ legnagyobb TCP/IP-hálózata. Ez az óra röviden áttekinti az Internet felépítését, valamint néhány fontosabb internetes szolgáltatást. Az Internet lesz a témája a következő két fejezetnek is, amelyekben a Világhálóval (World Wide Web, 17. óra) és az elektronikus levelezéssel (18. óra) foglalkozunk.

Az óra végeztével a következőkre leszünk képesek:

- Röviden le tudjuk írni az Internet felépítését.
- Felismerjük és le tudjuk írni egy egységes erőforrás-azonosító összetevőit.
- Az IPv4-címeket le tudjuk fordítani az IPv6 címterére.

Hogyan épül fel az Internet?

Nemigen találunk pontos leírást arról, hogy mi is az Internet. Az Internet legtöbb leírása sajnos az egyszerűséget részesíti előnyben a részletekkel szemben, így az olvasó alig kap többet annál a homályos képnél, hogy az Internet „az adatok országútja". Az Internet felépítése valójában olyan bonyolult, hogy csak kevés hivatásos hálózati rendszergazda képes pontosan megmondani, hogy mi is történik az Internetre kikerülő adatokkal. Persze nem is kell tudniuk: a TPC/IP stabilitása és sokoldalúsága lehetővé teszi, hogy egy adatcsomag belépjen az Internet ködébe, és a Föld túlsó oldalán pontosan a megfelelő helyen bukkanjon fel. De hová kerül az adatcsomag, amikor eltűnik a ködben?

Az Internet ma túl nagy ahhoz, hogy a 8.9. ábrán látott szerkezet, amelyben egyetlen gerinchálózat szolgál ki rendezetten elhelyezkedő önálló hálózatokat, képes legyen leírni. Az Internet jelenleg számos gerinchálózatból áll, amelyek közül sokat olyan magánvállalatok kezelnek, mint az AT&T, a Sprint vagy a Verizon.

Nem meglepő, hogy a Sprinthez és az AT&T-hez hasonló telefontársaságok főszerepet játszanak az Internet felépítésében. Ezeknek a távolsági telekommunikációs szolgáltatóknak a jelenléte is jól jelzi, hogy a telefonrendszerhez hasonlóan az Internet is sok-sok nagy távolságra kihúzott kábelből áll.

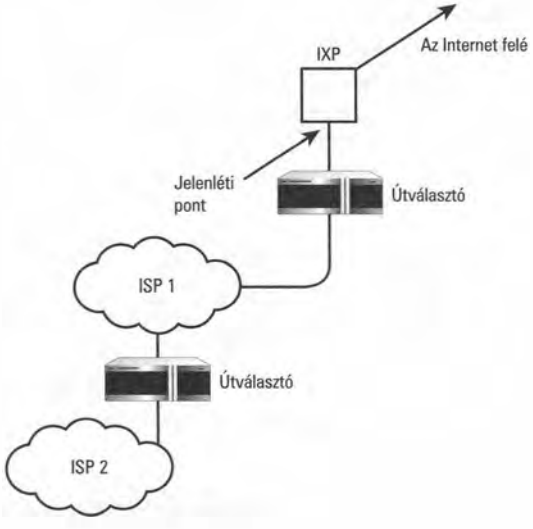
A gerinchálózatok nagy kapcsolótelepeken metszik egymást, amelyeknek *internetes cserepont* (Internet eXchange Point, IXP) a neve. Az Amerikai Egyesült Államokban a Verizon MAE East (Washington D.C.) és MAE West (San Jose, California) telepe a két legforgalmasabb internetes cserepont. Az IXP-k nagy telepek, ahol az internetszolgáltatók (Internet Service Provider, ISP) a hálózatukat az Internet gerinchálózatához kapcsolhatják. Az IXP-k nem nyújtanak útválasztási szolgáltatásokat – ezt az egyes internetszolgáltatók biztosítják, saját, az IXP-telep egy biztonságos területén elhelyezett útválasztóikon keresztül.

Az ISP-k *POP*- (Point of Presence, jelenléti pont) kapcsolatokat adnak bérbe (lásd a 16.1. ábrát). A nagy cserepont-telepekhez kapcsolódó ISP-k általában jelentős internetszolgáltatók. Egyesek közülük viszonteladóként működnek, és kisebb ISP-knek adnak bérbe sávszélességet, akik a vonalakat még kisebb szolgáltatóknak vagy cégeknek adhatják ki.

Az Internet tehát egymásba fonódó üzleti tranzakciók – lefedettség kiépítése, csatlakozás biztosítása a vonalak végén, sávszélesség bérbeadása – ezreiből áll, és sok ezer ISP nyújt szolgáltatásokat a felhasználóknak, vállalatoknak és intézményeknek. Így már megérthetjük, miért szokták az Internetet gyakran felhőként ábrázolni: messziről egyetlen tárgynak mutatja magát, de ahogy közelebb megyünk hozzá, sehol sem találjuk a közepét – mert nincs is neki: mindenütt körülvesz minket, akárhonnan nézzük.

Az, hogy az Internet mégis egyetlen egységes egészet alkot, nem a fizikai kapcsolatokból, hanem a következőkből ered:

- Általánosan érvényes szabályai vannak.
- Meghatározott intézmények gondoskodnak a fenntartásáról.
- Közös nyelvet beszél.



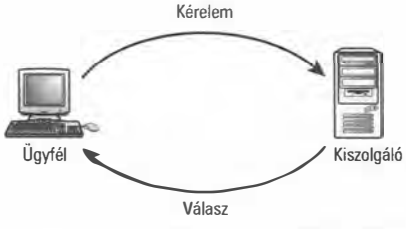
16.1. ábra
Az ISP-k jelenléti pontokat (POP) adnak bérbe az Interneten

Az 1. órán megismertük az Internetet irányító intézményeket, beleértve az IAB-t (Internet Advisory Board) és az IETF-et (Internet Engineering Task Force). Az Internet nyelve természetesen a TCP/IP, de érdemes külön kiemelni a TCP/IP rendszer egyik fontos elemét, amely globális szinten lehetővé teszi az üzenetküldést: a közös elnevezési és számozási rendszert, amelyet az ICANN felügyel. A DNS elnevezési rendszer több, mint a 11. órán megismert névfeloldási protokollok. A globális szintű névszolgáltatás hatalmas emberi erőfeszítést igényel azoknak az alacsonyabb szintű szervezeteknek az irányításához, amelyek az internetes nevek kiosztását felügyelik. A hatékony DNS elnevezési rendszer nélkül az Internet nem játszhatna olyan fontos szerepet a mindennapjainkban, mint ma.

Mi történik az Interneten?

Az Internet valójában egy hatalmas TCP/IP-hálózat. Ha nem aggódunk a biztonság vagy a késlekedés miatt, az Internetet szinte bármire használhatjuk, amit egy útválasztásos vállalati belső hálózaton megtehetünk. A biztonság ugyanakkor igen lényeges szempont. Az Internetet semmiképpen *nem szabad* olyasmire használni, amit egy útválasztásos vállalati belső hálózaton is megtehetünk – még ha ez lehetséges is. Annak az okait, hogy miért kell különösen ügyelnünk a biztonságra egy oyan védtelen területen, mint a nyílt Internet, a 22. és 23. fejezetben boncolgatjuk majd.

Fontos, hogy megjegyezzük, hogy minden hálózati tevékenységben részt vevő számítógépben (az Interneten vagy bármely más hálózaton) van egy közös dolog: olyan szoftvert futtatnak, amelyet az adott tevékenységre terveztek. A hálózat nem önmagától működik: protokollszoftver (például a 2-7. fejezetekben leírt TCP/IP-szoftver) szükséges hozzá, valamint alkalmazások a kapcsolat mindkét végén, amelyeket kifejezetten az egymással való kapcsolattartásra fejlesztettek ki. Ahogy a 16.2. ábra is mutatja, az Interneten a legtöbb számítógép vagy *ügyfélként* (szolgáltatásokat igénylő számítógépként), vagy *kiszolgálóként* (szolgáltatásokat nyújtó számítógépként) működik. Az ügyfélszámítógépeken futó ügyfélprogramokat kimondottan azzal a céllal írták, hogy együttműködjenek a kiszolgálókon futó kiszolgálóprogramokkal, a kiszolgálóprogramokat pedig arra szánták, hogy fogadják az ügyfelektől érkező kérelmeket, és válaszoljanak azokra.

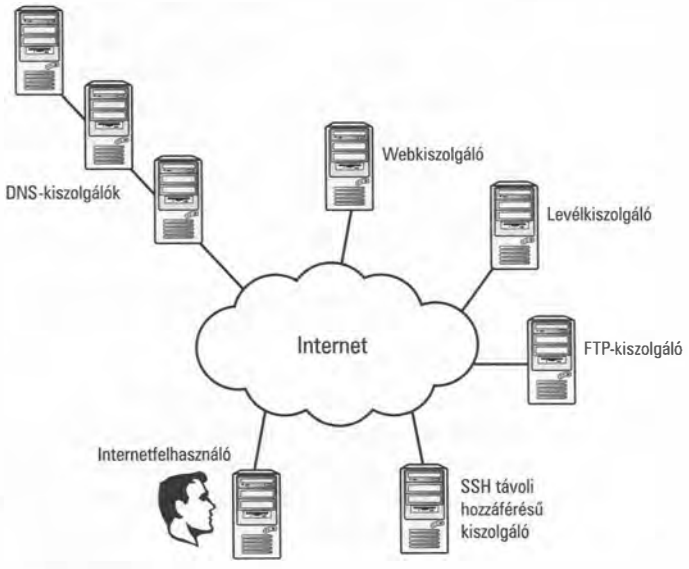


16.2. ábra

Az Interneten a számítógépek jellemzően vagy ügyfelként, vagy kiszolgálóként működnek

A 16.3. ábrán erre a sűrű ökoszisztémára vethetünk egy pillantást. A számítógépe előtt ülő felhasználó a világ bármely pontjáról tetszőleges földrajzi helyen található kiszolgálók ezreihez kapcsolódhat. Az egymással alá-fölérendelt viszonyban álló DNS-kiszolgálók a céltartomány nevét egy IP-címmé oldják fel (egy a felhasználó számára láthatatlan eljárás során), a felhasználó számítógépén működő ügyfélprogram pedig létrehozza a kapcsolatot. A kiszolgáló ezt követően weboldalakat jeleníthet meg, amelyek között a felhasználó böngészhet, azonnali üzenetküldést tehet lehetővé, FTP-n keresztül fájlok letöltését engedélyezheti, vagy ha a felhasználó esetleg egy levelezési kiszolgálóhoz kapcsolódott, letöltheti a beérkező üzeneteket a felhasználó gépére.

A szerény kezdetektől – néhány hálózatba kötött nagygéptől – az Internet eljutott odáig, hogy szolgáltatások olyan egyre terjeszkedő dzsungelévé vált, amelyről az Internetet eredetileg megalkotó professzorok és kutatók nem is álmodtak. A levélküldés és a webszörfölés mellett az internetfelhasználók új nemzedéke telefonálhat is, webkamerát üzemeltethet, tévét nézhet, zenét tölthet le, személyre szabott rádiót hallgathat, és webnaplóban teheti közzé a legféltettebb titkait – és mindezt a TCP/IP csodájának köszönhetően. (A későbbi órákon az új webes technológiák közül sokról szót ejtünk majd.)



16.3. ábra

Az Internet a Föld bármely pontjáról elérhető szolgáltatások hatalmas óceánja

URI-k és URL-ek

Ahogy a 16.3. ábra mutatja, az Internet erőforrásokat igénylő ügyfélrendszerek és erőforrásokat nyújtó kiszolgálók gigantikus masszája. Ha azonban közelebbről vizsgáljuk a működését, rájöhetünk, hogy a könyvben korábban tárgyalt protokollcímzési szabályok nem elegendőek ahhoz, hogy támogassák azt a rengeteg féle szolgáltatást, ami az Interneten elérhető. Az IP-cím vagy a tartománynév egy állomást (számítógépet) képes azonosítani. A kapuszám egy szolgáltatásra mutathat, ami az állomáson fut. De mit kér az ügyfél? Mit kell tennie a kiszolgálónak? Vannak bemenő adatok, amelyekre az ügyfél kimenő adatokat kér válaszként?

A szakemberek régen rájöttek annak a fontosságára, hogy az internetes erőforrások kérelmezésének legyen egy szabványos formája. Egyesek azt is érvként hozták fel, hogy az egységes kérelemformátum hozzájárulna ahhoz, hogy az Internet egyetlen nagy egésznek mutassa magát, nem pedig csak számítógépek összevisszaságának.

Az internetfelhasználók számára legismerősebb kérelemformátumot *egységes erőforráscímnek* (Uniform Resource Locator, URL) hívják. Az URL elsősorban a klasszikus webcímek formátumáról (például `http://www.mercurial.org`) ismert. Az URL-ek ma már annyira hétköznapinak számítanak, hogy a tévéreklámokban vagy a rágógumik csomagolásán sokszor mindenféle magyarázat nélkül szerepelnek.

Amire mi URL-ként gondolunk, az valójában egy általánosabb formátum, az *egységes erőforrás-azonosító* (Uniform Resource Identifier, URI) különleges esete. A két betűszót néha szinonimaként használják, de fontos, hogy megkülönböztessük őket. Az újabb internetes dokumentumok megpróbálják egymáshoz közelíteni a két szakkifejezést. Az RFC 3986 (Uniform Resource Identifier Generic Syntax) kimondja, hogy a jövőben a dokumentumok az URL helyett az általánosabb URI kifejezést fogják használni. Általános értelemben az *azonosító* (Identifier) pontosabb, mint a *cím* (Locator), mert nem minden kérelem mutat egy konkrét címre.

Az URI szerkezetének leírása 60 oldalnál is hosszabb, de az alapvető alak így fest:

```
séma://hatókör/útvonal?lekérdezés#töredék
```

A séma (scheme) a kérelem értelmezési rendszerét határozza meg. A séma mezőt gyakran egy protokollal azonosítják; az Interneten ma használatos sémák közül néhányat a 16.1. táblázatban mutatunk be. A webcímekben a klasszikus `http` sémát használják. Bár egyes sémák, például a `gopher`, ma már kevésbé fontosak, mint egykor voltak, más sémákat, például az `ftp`-t, ma is széles körben alkalmaznak.

A hatókör (authority), amelyet két perjel (`//`) vezet be, a kérelemhez tartozó felhasználót, állomást és kaput határozza meg. A hatókör elem teljes kifejezéssel valahogy így fest:

```
//joeyesterday@www.bonzai.com:8042
```

Ahogy a 6. órán megtanultuk, a protokollhoz általában tartozik egy alapértelmezett kapuszám, ezért a kapuszámot többnyire elhagyják. A felhasználónévre csak akkor van szükség, ha a felhasználónak azonosító adatokat kell megadnia az erőforrás eléréséhez, ami a Weben nem szokás, de az olyan protokollok esetében, mint az FTP, annál gyakoribb.

Előfordulhat, hogy akkor sem kell megadni egy felhasználót az URI-ban, ha a felhasználónak azonosítania kell magát. Sok szolgáltatás a kezdeti kérelem után kéri el a felhasználótól az azonosítóját és a jelszavát.

A felhasználó és a kapu nélkül a hatókör mező sokkal jobban hasonlít az általunk jól ismert egyszerű webcímekre:

```
//www.bonzai.com
```

Vagy, ha a séma elemet is hozzáadjuk:

```
http://www.bonzai.com
```

Ebben a példában az állomást egy DNS-tartománynévvel adtuk meg, de a számítógépekre az IP-címükkel is hivatkozhatunk.

Az útvonal (path) elem egymásba ágyazott könyvtárakon keresztül arra a fájlra mutat, amely a kérelem tárgya. A `http` esetében, ha az útvonalat elhagyjuk, a kérelem a tartomány alapértelmezett weboldalára (a honlapra vagy kezdőlapra) mutat. A legtöbb felhasználó már jól tudja, hogy a tartománynév után további könyvtár- és fájlneveket kell megadni:

```
http://www.bonzai.com/trees/LittleTrees.pdf
```

Az URI lekérdezés (query) és töredék (fragment) elemeit ritkán írja be vagy értelmezi ember. Ezeknek az összetevőknek a pontos jelentése a sémától függ, és egyes sémák nem is támogatják őket. A „vadonban" akkor figyelhetünk meg egy lekérdezésmezőt, ha egy olyan keresőprogramba, mint a Google, egy keresési kérelmet írunk be, majd megvizsgáljuk a címsávban megjelenő URI-t.

A fenti példa az URI-t a Világhálón használt, rendkívül népszerű HTTP protokoll környezetében mutatta be. (A HTTP-ről és a hozzá társuló leírónyelvről, a HTML-ről a 17. órán beszélünk bővebben.) Tartsuk ugyanakkor észben, hogy a különböző sémaszabványok más-más módon határozhatják meg az URI-ban található információk értelmezését. Az URI általános leírását szándékosan választották külön az egyes sémaszabványokban leírt részletektől – így a sémák az alapvető formátum módosításának igénye nélkül fejleszthetők tovább. A 16.1. táblázatban az egyes sémákhoz tartozó RFC-dokumentumokat is felsoroltuk.

16.1. táblázat *URI-sémák*

Séma	Leírás	Hivatkozás
`file`	Fájl a gazdarendszeren	RFC 1738
`ftp`	Fájlátvitelli protokoll	RFC 1738
`gopher`	A Gopher protokoll	RFC 4266
`http`	Hiperszöveg-átviteli protokoll	RFC 2616
`https`	Biztonságos hiperszöveg-átviteli protokoll	RFC 2818
`im`	Azonnal üzenetküldés	RFC 3860
`ldap`	Pehelysúlyú könyvtárelérési protokoll	RFC 4516
`mailto`	Elektronikuslevél-cím	RFC 2368
`nfs`	Hálózati fájlrendszer protokoll	RFC 2224
`pop`	Postahivatal protokoll 3. változat	RFC 2384
`telnet`	Interaktív Telnet-munkamenet	RFC 4248

Összefoglalás

Az Internet szerte a világon elhelyezkedő számítógépekből áll, amelyek szolgáltatásokat nyújtanak és igényelnek. Ezeknek az erőforrásoknak az azonosítására és megcímzésére az URI formátum biztosít szabványos alakot. A protokollok ugyanakkor eltérnek egymástól, és a kommunikáció részletei is a szolgáltatástól függően változhatnak. Az Interneten ma használatos legfontosabb szolgáltatások közül a későbbi fejezetekben mutatunk be néhányat.

Kérdezz-felelek

K *A cégem internetszolgáltatóvá (ISP-vé) szeretne válni. Megpróbáltunk POP-kapcsolatot létesíteni egy közeli NAP-vel, de nem volt szabad hely. Hogyan kapcsolódhatnánk az Internethez?*

V Igényeljünk sávszélességet egy viszonteladó ISP-től.

K *Miért akar néhány ázsiai és kelet-európai ország saját formátumot alkalmazni a DNS és az URI helyett?*

V A Latin karakterkészlet korlátai akadályozzák azokat a felhasználókat, akik nem latin betűkkel írt nyelven beszélnek.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Egységes erőforrás-azonosító (URI)** – Alfanumerikus karakterlánc, amely egy internetes erőforrás azonosítására szolgál.
- **Egységes erőforráscím (URL)** – Az URI egyik típusa, amely egy erőforrás címét adja meg. Az egyik leggyakrabban használt URL-alakot a webcímek (például `www.sams.com`) jelentik.
- **Hatókör** – Az URI-nak az állomást, a felhasználót és a kaput azonosító része.
- **Internetes cserepont (IXP)** – Az Internethez hozzáférést nyújtó telep.
- **Jelenléti pont (POP)** – Kapcsolódási pont az Internethez, amelyet egy ISP ad bérbe.
- **Séma** – Az URI-nak az a része, amely meghatározza az URI fennmaradó részének értelmezésére használandó protokollt vagy rendszert.

17. ÓRA

A HTTP, a HTML és a Világháló

A fejezet tartalmából:

- HTML
- HTTP

A Világháló (World Wide Web) az Internet univerzális grafikus megjelenítési keretrendszereként kezdte a pályafutását. A Web a fogantatása óta az Internetet jelenti a nagyközönség szemében, és forradalmasította az alkalmazások felületével kapcsolatos szemléletünket. Ez az óra a HTTP, a HTML és a Web világába nyújt bevezetést.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni a Világháló működését.
- Fel tudunk építeni egy alapszintű weboldalt szöveg és HTML-címkék segítségével.
- El tudjuk magyarázni a HTTP protokoll működését.

Mi a Világháló?

A weboldalak képe, amit a böngészőnk ablakában látunk, a böngésző és a webkiszolgáló számítógép közötti társalgás eredménye. Ennek a társalgásnak a nyelve a HTTP (Hypertext Transfer Protocol, hiperszöveg-átviteli protokoll). A kiszolgálótól az ügyfélhez érkező adattartalom szövegek, képek, címek és formázókódok finoman szerkesztett szövevénye, amelyből egy hihetetlenül sokoldalú formázónyelv, a HTML (Hypertext Markup Language, hiperszöveges jelölőnyelv) állít elő egységes kinézetű dokumentumot.

Annak a valaminek az alapelemeit, amit ma Világhálóként (World Wide Web) ismerünk, Tim Berners-Lee alkotta meg 1989-ben, a svájci Genf CERN kutatóintézetében. Berners-Lee egy kifinomult és hatékony információs rendszert hozott létre három olyan technológia vegyítésével, amely akkoriban már fejlesztés alatt állt:

- **Jelölőnyelv (leírónyelv)** – Utasítások és formázókódok rendszere, amelyeket szövegbe ágyaznak.
- **Hiperszöveg** – A dokumentumokra, képekre és más elemekre mutató hivatkozások szövegbe való beágyazását lehetővé tevő rendszer.
- **Az Internet** – (ahogy ma ismerjük) Globális számítógép-hálózat, amely szolgáltatásokat igénylő ügyfelekből és a TCP/IP-n keresztül szolgáltatásokat nyújtó kiszolgálókból áll.

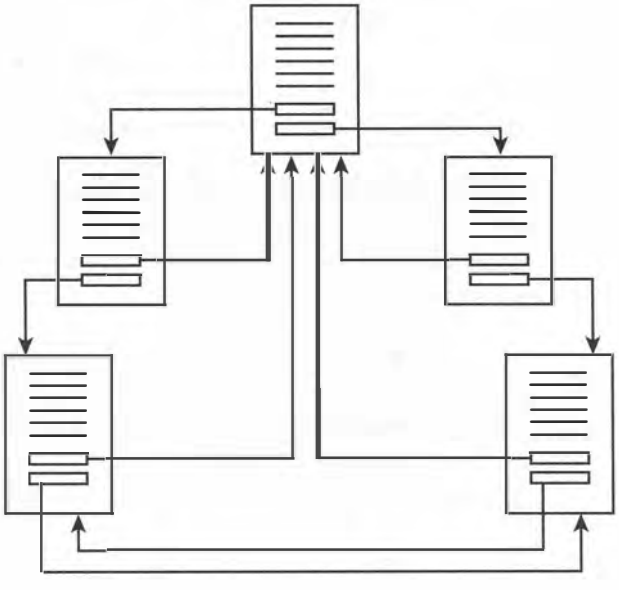
A jelölőnyelvek az 1960-as években kezdték a pályafutásukat, és arra használták őket, hogy formázó- és tördelőkódokat adjanak a korai számítógépek által használt egyszerű szövegekhez. Abban az időben a számítógépes világban a beállítófájlok, az elektronikus súgódokumentumok és az elektronikus levelek egyszerű szövegfájlok voltak. Amikor azonban a számítógépeket levelezésre, illetve jegyzetek és más „kész" dokumentumok készítésére kezdték használni, szükség volt valamilyen módszerre, amivel olyan elemeket lehet meghatározni, mint a címsorok, a dőlt és félkövér betűk vagy a margók. Az első jelölőnyelvek némelyikét (például a ma is használatban levő TeX-et) tudósok számára fejlesztették ki, hogy segítsék a matematikai egyenletek formázását és tördelését.

Mire megjelentek a modern szövegszerkesztő programok, a gyártók számos (sokszor jogvédett) rendszert dolgoztak ki a formázási információk szöveges dokumentumokba való bekódolására. E rendszerek közül néhány ASCII alapú kódokat alkalmazott, míg mások különféle digitális jelzésekkel jelölték a formázási információkat.

Természetesen ezek a formázókód-rendszerek csak akkor működnek, ha a dokumentumot író és az azt olvasó alkalmazás megegyezik az egyes kódok jelentésében.

Berners-Lee és a HTML más úttörői egy univerzális, gyártósemleges rendszert szerettek volna a formázási információk kódolására. Azt akarták, hogy ez a leírórendszer ne csak tördelési kódokat tartalmazzon, hanem hivatkozásokat is képfájlokra és más dokumentumokra.

A hiperszöveg (a szövegbe ágyazott „élő" hivatkozások, amelyek megjelenítik a hivatkozott dokumentumot) ötlete szintén az 1960-as években merült fel. Berners-Lee az URL (vagy URI – lásd a 16. órát) kifejlesztésén keresztül emelte be a hiperszöveg ötletét az Internet szerkezetébe. A hivatkozások (link) lehetővé teszik, hogy az olvasó kis adagokban tekintse meg a hálózaton levő információkat. Az olvasó választhat, hogy a hivatkozást követve új oldalra ugrik-e, ahol további információkhoz juthat. A HTML-dokumentumokból oldalak és hivatkozások egységes rendszerei építhetők fel (lásd a 17.1. ábrát). A látogatók a hivatkozások bejárási sorrendjétől függően más-más útvonalon böngészhetnek az adatok között, a webfejlesztőnek pedig szinte korlátlan lehetőségei vannak annak meghatározására, hogy hová vezessenek az egyes hivatkozások. Egy hivatkozás vezethet ugyanannak a könyvtárnak egy másik HTML-dokumentumához, de egy másik könyvtárban, sőt akár egy másik számítógépen levő dokumentumhoz is. A hivatkozáson keresztül tehát egy teljesen eltérő, a világ túlsó oldalán található számítógépen elhelyezett webhelyre is kerülhetünk.



17.1. ábra
A webhelyek oldalak és hivatkozások egységes rendszerei

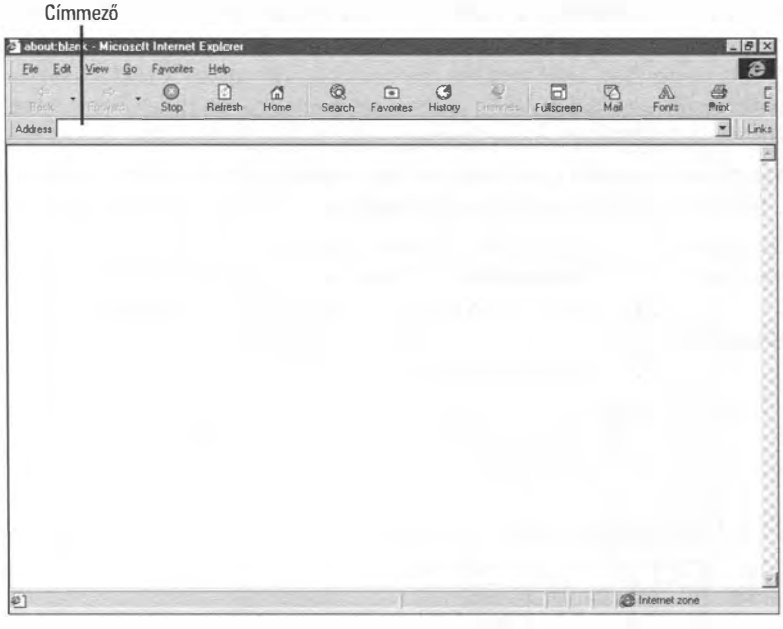
Ahogy a 16. órán megtanultuk, az URL-nek az az alakja, amelyet a Webről a legjobban ismerünk, így fest:

```
http://www.dobro.com
```

Gyakran láthatunk az URL-hez hozzáfűzve egy elérési utat és egy fájlnevet is:

```
http://www.dobro.com/techniques/repair/fix.html
```

A webböngésző programok az URL-ek segítségével közlekednek. Egy weboldalt úgy érhetünk el, hogy beírjuk az oldal URL-jét a böngésző ablakának címmezőjébe (lásd a 17.2. ábrát). Amikor egy hivatkozásra kattintunk, a böngésző megnyitja a hivatkozás URL-jében meghatározott weboldalt.



17.2. ábra
Írjuk be az URL-t a böngésző ablakának címmezőjébe

E rövid bevezető összefoglalásául vegyük sorra, hogy milyen elemek kombinációjából épülhet fel egy alapszintű HTML-dokumentum:

- Szöveg
- Képek
- Szövegformázó kódok (a betűkre és az elrendezésre vonatkozó információk)
- Másodlagos fájlokra (például képekre) mutató hivatkozások
- Más HTML-dokumentumokra vagy az adott dokumentum más helyeire mutató hivatkozások

Ha a felhasználó el szeretne látogatni egy webhelyre, akkor beírja a webhely URL-jét a webböngésző ablakába. A böngésző ekkor kapcsolatot létesít az URL-ben meghatározott webkiszolgálóval, a kiszolgáló a hálózaton át elküldi a HTML-adatokat a böngészőnek, a böngésző pedig a HTML-adatokat értelmezve létrehozza a weboldal nézetét a böngésző ablakában.

A HTML működése

A HTML az a hasznos tartalom, amelyet a HTTP-n keresztül átviszünk. Ahogy az óra korábbi részében megtanultuk, a HTML-dokumentumok szöveget, formázókódokat, valamint más fájlokra mutató hivatkozásokat tartalmaznak. Ha megvizsgáljuk egy alapszintű HTML-dokumentum tartalmát egy olyan egyszerű szövegszerkesztő alkalmazásban, mint a Windows Jegyzettömbje (Notepad) vagy a Unix `vi` programja, azt láthatjuk, hogy a dokumentum valójában egy közönséges szövegfájl. A fájl tartalmazza az oldalon megjelenő szöveget, de emellett különleges HTML-kódokat is, amelyeket *címkéknek* (tag) neveznek. A címkék a böngészőnek szóló utasítások. Nem jelennek meg a weboldalon, az adatok megjelenésére és az oldal viselkedésére azonban hatással vannak. A HTML-címkék teszik lehetővé a formázást, illetve a fájlokra mutató hivatkozások használatát a weboldalon. A legfontosabb HTML-címkéket a 17.1. táblázatban soroltuk fel.

17.1. táblázat *Néhány fontos HTML-címke*

Címke	Leírás
`<HTML>`	A fájl HTML-tartalmának elejét és végét jelzi.
`<HEAD>`	A fejrész elejét és végét jelzi.
`<BODY>`	A böngésző ablakában megjelenő szöveget leíró törzs elejét és végét jelzi.
`<H1>`, `<H2>`, `<H3>`, `<H4>`, `<H5>` és `<H6>`	Egy címsor elejét és végét jelzik. Az egyes címsorcímkék más-más szintű címsort jelképeznek; a legmagasabb szint a `<H1>`.
`<B>`	Egy félkövérrel írt szövegrész elejét és végét jelzi.
`<U>`	Egy aláhúzott szövegrész elejét és végét jelzi.
`<I>`	Egy dőlt betűs szövegrész elejét és végét jelzi.
`<FONT>`	Egy adott szövegjellemzőkkel kiírandó szövegrész elejét és végét jelzi. A rendelkezésre álló szövegjellemzők közül a 17.2. táblázatban soroltunk fel néhányat.
`<A>`	Egy horgonyt határoz meg, amit általában egy hivatkozás jelzésére használnak. A hivatkozás céljának URL-je az első `<A>` címke belsejében szerepel, a `HREF` jellemző értékeként (lásd ebben a részben valamivel lejjebb).
`<IMG>`	Egy olyan képfájlt határoz meg, amelyet a szöveg belsejében kell megjeleníteni. A képfájl URL-je a címkében az `SRC` jellemző értékeként szerepel. (A jellemzőkről ebben a részben bővebben is szó lesz majd.)

A HTML természetesen sokkal több elemből áll, mint amit egyetlen táblázatban fel tudnánk sorolni. Sok címke szövegblokkokra vonatkozik; ebben az esetben a címke a blokk elején és végén szerepel. A blokk végét jelző címkében a perjel (/) jelzi, hogy záró címkéről van szó. Más szavakkal, egy H1 szintű címsort például így kell felcímkéznünk:

```
<H1>Dewey Defeats Truman</H1>
```

A HTML-dokumentumoknak elvileg a dokumentumtípus-meghatározással (<!DOCTYPE>) kell kezdődniük. A !DOCTYPE a dokumentumban használt HTML-változatot határozza meg; a HTML 4.0 esetében például így:

```
<!DOCTYPE HTML PUBLIC " -//W3C/DTD HTML 4.0//EN">
```

(A különleges böngészőbővítményeket használó weboldalak más dokumentumtípust is meghatározhatnak.)

A legtöbb böngésző ugyanakkor nem igényli a !DOCTYPE utasítást, ezért sok HTML-oktatókönyv nem is foglalkozik vele.

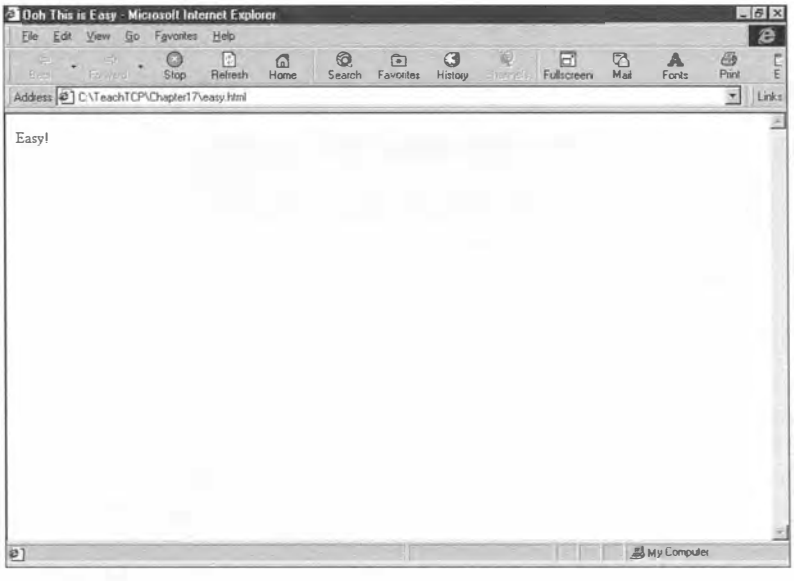
A !DOCTYPE utasítást a <HTML> címke követi. A dokumentum fennmaradó része a <HTML> és a hozzá tartozó, a fájlt lezáró </HTML> címke között áll. A kezdő és záró <HTML> címke által közrefogott dokumentumrész az alábbi két részre oszlik:

- A fej (amely a <HEAD> és a </HEAD> címke között áll) a dokumentumról tartalmaz információkat. A fejrészben szereplő információk nem jelennek meg a weboldalon, bár a <TITLE> címke által meghatározott címet a böngészőablak címsorában láthatjuk. A <TITLE> kötelező elem, a <HEAD> rész egyes elemei azonban elhagyhatók – ilyen például a dokumentumban használt stílusokat leíró <STYLE>. Ha a <STYLE> elemről többet szeretnénk tudni, olvassunk el egy könyvet vagy cikket a HTML-ről.
- A törzs (amelyet a <BODY> és </BODY> címkék zárnak közre) a weboldalon ténylegesen megjelenő szöveget és a hozzá kapcsolódó HTML-címkéket tartalmazza.

Következzék egy egyszerű HTML-dokumentum:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE> Ooh This is Easy </TITLE>
</HEAD>
<BODY>
Easy!
</BODY>
<HTML>
```

Ha ezt a HTML-kódot egy szövegfájlba mentjük, majd a fájlt megnyitjuk egy webböngészőben, az `Easy!` szöveg jelenik meg a böngésző ablakában. (A böngészőnktől és az operációs rendszertől függően előfordulhat, hogy a fájlt `.htm` vagy `.html` kiterjesztéssel kell mentenünk, vagy HTML-fájlként kell megnyitnunk.) A címsorban az `Ooh This is Easy` szöveget láthatjuk (lásd a 17.3. ábrát).



17.3. ábra

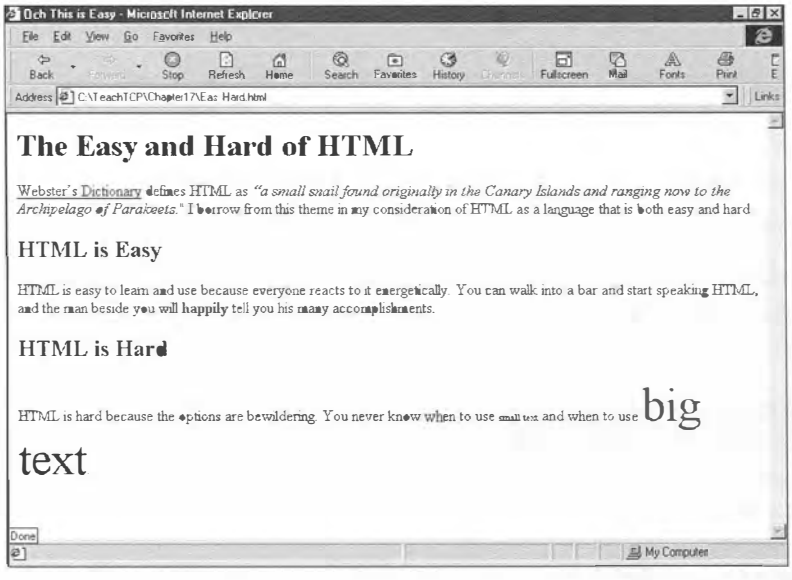
Egy igen egyszerű weboldal

Az oldalt a törzsrészben további szöveggel és formázással fűszerezhetjük. Az alábbi példában `<H1>` és `<H2>` szintű címsorokat adunk az oldalhoz, a `<P>` címkével egy bekezdésnyi szöveget, a `<B>` címkével félkövér, az `<I>` címkével dőlt betűket, a `<FONT>` címkével pedig betűméretet állítunk be. Figyeljük meg, hogy a `<FONT>` címkében egy jellemző is szerepel. A jellemzők (attribútumok) a címkék belsejébe zárt paraméterek, amelyek kiegészítő információt nyújtanak. A 17.2. táblázatban további szövegjellemzőket láthatunk.


```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE> Ooh This is Easy </TITLE>
</HEAD>
<BODY>
<H1>The Easy and Hard of HTML</H1>
<P><U>Webster's Dictionary</U> defines HTML as <I>"a small snail found
originally in the Archipelago of Parakeets." I borrow from this theme in
my consideration of HTML.</P><H2>HTML is Easy</H2>
<P>HTML is easy to learn and use because everyone reacts to it
```

```
energetically. You can walk into a bar and start speaking HTML, and the
man beside you will <B>happily</B> tell you his many
accomplishments.</P>
<H2>HTML is Hard</H2>
<P>HTML is hard because the options are bewildering. You never know when
to use <FONT SIZE=1>small text</FONT> and when to use <FONT SIZE=7>big
 text</FONT>.</P>
</BODY>
</HTML>
```

Ez a kód a böngészőben úgy jelenik meg, ahogy a 17.4. ábrán láthatjuk.



17.4. ábra
Az egyszerű példa bővítése

17.2. táblázat *A <FONT> HTML-címke jellemzői*

Jellemző	Leírás
SIZE	Viszonyított (relatív) betűméretet állít be. Az értéke 1-től 7-ig terjedhet (például: <FONT SIZE=7>).
LANG	Annak a nyelvnek a kódját adja meg, amelyen a szöveget írták.
FACE	A betűtípust határozza meg. Például: <FONT FACE="Arial">.
COLOR	A szöveg színét állítja be. Például: <FONT COLOR="RED">.

Ahogy az óra korábbi részében megtanultuk, a hiperszöveges hivatkozások (hiperhivatkozások) a webtervezés fontos elemei. A hivatkozások más dokumentumokra vagy az adott dokumentum egy másik részére mutathatnak. Ha a felhasználó egy hivatkozás kiemelt szövegére kattint, a böngésző azonnal megnyitja a hivatkozott dokumentumot. Ez azt a hatást kelti, mintha a felhasználó színes és informatív tartalmak végtelen kertjében sétálna.

Amikor ebben a virágzó kertben sétálunk, időnként álljunk meg, és jusson az eszünkbe, hogy a böngésző angol megfelelője, a *browser* kifejezés eredetileg egy zsiráfot vagy nagy dinoszauruszt jelentett, ahogy a fákról csócsálja a leveleket.

A hivatkozások a HTML-fájlban címkeként jelennek meg. Egy hivatkozás a legegyszerűbb formájában az `<A>` címkét használja, a hivatkozás céljának URL-jét a címke `HREF` jellemzőjének értékeként megadva. A fenti kódban például, ha azt szeretnénk, hogy az „Archipelago of Parakeets" szavak hiperhivatkozásként jelenjenek meg, ami az említett szigetvilágról szóló webhelyre vezet, zárjuk `<A>` címkék közé ezt a szövegrészt a következőképpen:

```
originally in the <A HREF="http://www.ArchipelagoParakeets.com">
Archipelago of Parakeets</A>. I borrow from this theme
```

A sokoldalú HTML formátum számos további lehetőséget biztosít. Hivatkozást elhelyezhetünk például egy kép belsejében is; saját stíluslapokat hozhatunk létre, amelyekben különleges címkék határozzák meg a bekezdések stílusát; a weboldalt táblázatokra, hasábokra, űrlapokra és keretekre tagolhatjuk; vagy választógombokat, jelölőnégyzeteket és lenyíló listákat adhatunk hozzá. A HTML elterjedésének elején a tervezők minden HTML-kódot szövegszerkesztővel kódoltak közvetlenül a dokumentumokba (ahogy az előző példákban mi is tettük), a mai profi webtervezők azonban már kifejezetten webfejlesztésre szánt alkalmazásokkal – ilyen például az Adobe Dreamweaver vagy a Microsoft FrontPage – dolgoznak, amelyek elrejtik a HTML részleteit, és a tervezőnek úgy mutatják az oldalt, ahogy majd a felhasználók is látni fogják. Az erőfeszítés nélküli webtervezéshez emellett olyan új eszközök nyújtanak további lehetőségeket, mint a wikik, illetve a tartalomkezelő rendszerek (CMS, Content Management System).

Még ma is széles körben használnak olyan statikus, előre formázott HTML-dokumentumokat, mint amilyeneket ebben a részben láttunk, sok webhely azonban a dinamikus HTML segítségével csak akkor állítja elő a webes tartalmat, amikor egy ügyfél kéri azt.

A klasszikus HTML-címkékben a kis- vagy nagybetűk használata nem számít, az olyan későbbi szabványok, mint az XML és az XHTML, azonban nagyobb figyelmet fordítanak erre. Az XML megkülönbözteti a kis- és nagybetűket, az XHTML pedig kisbetűs elem- és jellemzőneveket követel meg.

A HTTP működése

Ahogy korábban megtanultuk, a webkiszolgálók és a böngészők a HTTP-n (Hypertext Transfer Protocol, hiperszöveg-átviteli protokoll) keresztül társalognak egymással. A HTTP (1.1) leírását az RFC 2616 tartalmazza, a későbbi dokumentumok pedig bővítik a HTTP képességeit. A HTTP célja a HTML-dokumentumok átvitelének támogatása. A HTTP alkalmazásszintű protokoll. A HTTP-ügyfél és -kiszolgálóprogramok a megbízható TCP szállítási protokollt használják a kapcsolatok létesítésére.

A HTTP a következő feladatokat látja el:

- Kapcsolatot létesít a böngésző (az ügyfél) és a kiszolgáló között.
- Egyezteti a beállításokat, és beállítja a paramétereket a munkamenet számára.
- Biztosítja a HTML-tartalom megfelelő átvitelét.
- Bontja a kapcsolatot a kiszolgálóval.

Bár a webes kommunikáció természete ma már rendkívül bonyolult, ez a bonyolultság nagyrészt abból ered, ahogy a kiszolgáló felépíti a HTML-tartalmat, illetve amit a böngésző csinál a kapott tartalommal. Magának a tartalomnak az átvitele a HTTP-n keresztül viszonylag egyszerű.

Amikor beírunk egy URL-t a böngészőablakba, a böngésző először megvizsgálja az URL sémáját, hogy meghatározza a protokollt. (A legtöbb webböngésző a HTTP mellett más protokollokat is támogat.) Ha a böngésző azt állapítja meg, hogy az URL egy HTTP-webhelyen található erőforrásra mutat, akkor kinyeri a DNS-nevet az URL-ből, és névfeloldási eljárást kezdeményez. Az ügyfélgép elküldi a DNS-keresési kérelmet egy névkiszolgálónak, és megkapja a kiszolgáló IP-címét. A böngésző ezt követően a kiszolgáló IP-címével TCP-kapcsolatot kezdeményez a kiszolgálóval (a TCP-ről a 6. órán beszéltünk bővebben).

A HTTP régebbi (az 1.1-es előtti) változataiban az ügyfél és a kiszolgáló minden átvitt egységhez új TCP-kapcsolatot nyitott. A HTTP újabb változatai már lehetővé teszik, hogy az ügyfél és a kiszolgáló maradandó kapcsolatot tartsanak fenn.

Miután a TCP-kapcsolat létrejött, a böngésző a HTTP `GET` parancsával lekéri a weboldalt a kiszolgálóról. A `GET` parancs a böngésző által kért erőforrás URL-jét tartalmazza, illetve a tranzakcióhoz használni kívánt HTTP-változatot. A legtöbb esetben a böngészőnek elég egy viszonyított (relatív) URL-t megadnia a `GET`-kérelemben (tehát nem a teljes URL-t), mert a kapcsolat a kiszolgálóval ekkor már létrejött:

```
GET /watergate/tapes/transcript HTTP/1.1
```

A `GET` parancsot egyéb nem kötelező *mező:érték* párok követhetik, amelyek olyan beállításokat határoznak meg, mint a nyelv, a böngésző típusa vagy az elfogadható fájltípusok.

A kiszolgáló válasza egy fejlécből áll, amelyet a kért dokumentum követ. A válaszfejléc alakja így fest:

```
HTTP/1.1 állapotkód indoklás
mező:érték
mező:érték...
```

Az állapotkód egy háromjegyű szám, amely a kérelem állapotát írja le, az indoklás pedig az állapot rövid leírása. Néhány gyakori állapotkódot a 17.3. táblázatban soroltunk fel. Amint láthatjuk, az állapotkód bal szélső számjegye egy általános kategóriát határoz meg: a 100-as állapotkódok információt nyújtanak, a 200-asok sikert jeleznek, a 300-asok átirányítást, a 400-asok ügyfélhibát, az 500-asok pedig kiszolgálóhibát. A hírhedt 404-es kód már ismerős lehet, mert gyakran ez jelenik meg válaszként, ha nem létező oldalt próbálunk megnyitni, vagy elírtuk az URL-t. Az ügyfél kérelméhez hasonlóan a kiszolgáló válasza is tartalmazhat további nem kötelező *mező*:*érték* párokat. A fejlécmezők közül a 17.4. táblázat mutat néhányat. A böngésző minden olyan mezőt figyelmen kívül hagy, amelyet nem tud értelmezni.

17.3. táblázat *Néhány gyakran előforduló HTTP-állapotkód*

Kód	Indoklás	Leírás
100	Continue (Folytatás)	A kérelem feldolgozása folyamatban van.
200	OK	A kérelem sikeres volt.
202	Accepted (Elfogadva)	A kérelem feldolgozásra elfogadva, de a feldolgozás még tart.
301	Moving Permanently (Véglegesen áthelyezve)	Az erőforrásnak új címe van.
302	Moving Temporarily (Ideiglenesen áthelyezve)	Az erőforrásnak új ideiglenes címe van.
400	Bad Request (Hibás kérelem)	A kiszolgáló nem ismeri fel a kérelmet.
401	Unauthorized (Jogosulatlan hozzáférés)	A hitelesítés meghiúsult.
404	Not Found (Nem található)	A kért erőforrás nem létezik.
406	Not Acceptable (Nem elfogadható)	A tartalom nem elfogadható a böngésző számára.
500	Internal Server Error (Belső kiszolgálóhiba)	A kiszolgáló hibát észlelt.
503	Service Unavailable (A szolgáltatás nem érhető el)	A kiszolgáló túlterhelt vagy nem működik.

17.4. táblázat *Példák HTTP-fejlécmezőkre*

Mező	Az érték kötelező típusa	Leírás
Content-Length	integer	A tartalomobjektum mérete oktettben
Content-Encoding	x-compress x-gzip	Az üzenet kódolásának típusát jelző érték

17.4. táblázat *Példák HTTP-fejlécmezőkre*

Mező	Az érték kötelező típusa	Leírás
`Date`	Az RFC 850-ben meghatározott szabványos dátumformátum	Az objektum létrehozásának ideje greenwich-i középidő (GMT) szerint
`Last-modified date`	Az RFC 850-ben meghatározott szabványos dátumformátum	Az objektum utolsó módosításának ideje greenwich-i középidő (GMT) szerint
`Content-Language`	Nyelvi kód az ISO 3316 szerint	A nyelv, amelyen az objektumot írták

Ahogy a 17.4. táblázatból láthatjuk, egyes fejlécmezők csupán információt nyújtanak, míg más mezők a beérkező HTML-dokumentum értelmezéséhez és feldolgozásához szükséges adatokat tartalmaznak.

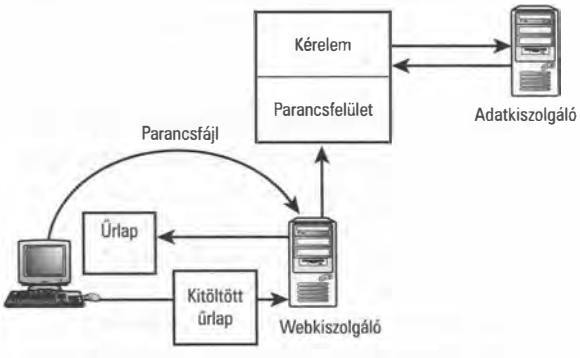
A fejlécmezőknek a HTML-ben használatos formátumát az elektronikus leveleknek az RFC 822-ben meghatározott fejlécformátumától kölcsönözték.

A `Content-Length` mező különösen fontos. A HTTP korábbi, 1.0-s változatában minden kérelem–válasz ciklus új TCP-kapcsolatot igényelt. Az ügyfél megnyitott egy kapcsolatot, és elindította a kérelmet, amit a kiszolgáló teljesített, majd bezárta a kapcsolatot. Ebben a rendszerben az ügyfél tudta, hogy a kiszolgáló mikor fejezte be az adatok elküldését – akkor, amikor a kiszolgáló bontotta a TCP-kapcsolatot. Sajnos azonban ez az eljárás a kapcsolatok folyamatos nyitogatása és bezárása miatt többletterhelést jelentett, ezért a HTTP 1.1 lehetővé tette, hogy az ügyfél és a kiszolgáló egyetlen átvitelnél hosszabb ideig is fenntarthassa a kapcsolatot. Így viszont az ügyfélnek valahogyan tudomást kell szereznie arról, hogy mikor fejeződött be egy adott válasz. A `Content-Length` mező határozza meg a válaszhoz kapcsolódó HTML-objektum hosszát. Ha a kiszolgáló nem ismeri az általa elküldött objektum hosszát – ami a dinamikus HTML megjelenésével egyre gyakoribb –, a `Connection:close` fejlécmezőt küldi el, hogy értesítse a böngészőt, hogy az adatok végét úgy fogja jelezni, hogy bezárja a kapcsolatot.

A HTTP ezenkívül egy egyeztetési szakaszt is támogat, amelynek során a kiszolgáló és az ügyfél megegyeznek bizonyos formátumokban és egyéb beállításokban.

Dinamikus HTML

A Web eredetileg egyszerű, statikus szövegfájlokként tekintett a HTML-fájlokra, amelyeket minden kérelem esetében egységesen kell szolgáltatni, de az utóbbi években a webes technológiák fejlődésének köszönhetően a helyzet bonyolultabbá vált. A webhelyek ma már gyakran akkor állítják elő a tartalmat, amikor az ügyfél kérelme beérkezik. Ezek a dinamikus HTML-eljárások lehetővé teszik, hogy a tartalom a felhasználó igényeihez igazodjon, emellett a dinamikus HTML a webtervezést is egyszerűbbé teszi (ha sikerrel vettük a programozási akadályokat), mert a webkiszolgáló egyetlen sablonra alapozva kimenetek korlátlan számú kombinációját állíthatja elő.



17.5. ábra

Példa a kiszolgálóoldali parancsfájlok működésére

Egy számítógépprogramot vagy parancsfájlt meglehetősen egyszerű rávenni, hogy HTML-tartalmat állítson elő. Ez a dinamikus megközelítés lehetővé teszi, hogy a webhelyek interaktív kapcsolatot alakítsanak ki a felhasználóval, mert a kiszolgáló megteheti, hogy egy weboldalt a felhasználótól kapott bemenetnek megfelelően állítson össze. A kiszolgálóoldali parancsfájloknak köszönhetően a kiszolgáló bemenetet fogadhat az ügyféltől, és ezt a bemenetet a színfalak mögött dolgozhatja fel.
A kiszolgálóoldali parancsfájlok használatára a 17.5. ábrán láthatunk egy szokványos forgatókönyvet. Az eljárás a következőképpen zajlik:

1. A felhasználó megnyit egy oldalt, amelyen egy űrlap található, például egy termék megvásárlásához vagy a látogató adatainak felvételéhez.
2. A kiszolgáló előállítja az űrlapot a felhasználó választásai alapján, és elküldi a böngészőnek.
3. A felhasználó beírja a szükséges adatokat az űrlapra, és a böngésző visszaküldi az űrlapot a kiszolgálónak. (Vegyük észre, hogy a HTML-űrlapok megfordítják a szokásos eljárást: a böngésző küld tartalmat a kiszolgálónak a kiszolgáló kérésére.)

4. A kiszolgáló fogadja az adatokat a böngészőtől, és egy programozási felületen keresztül átadja azokat azoknak a programoknak, amelyek feldolgozzák a felhasználótól kapott információkat. Amennyiben a felhasználó egy terméket vásárol, ezek a háttérben működő programok ellenőrizhetik például a hitelkártya-adatokat, vagy rögzíthetik a rendelést, ha pedig a felhasználó egy levelezőlistára iratkozik fel, vagy egy korlátozott hozzáférésű webhelyhez csatlakozik, felvehetik a felhasználó adatait egy adatbázisba.

Számos programozási nyelv és környezet született, hogy segítsen a fejlesztőknek a kiszolgáló alapú webalkalmazások felépítésében. Az egyik módszer egy weboldal és egy program vagy parancsfájl érintkezési felületének biztosítására a CGI (Common Gateway Interface, közös átjárófelület) használata. A CGI-t arra a célra fejlesztették ki, hogy űrlapokról származó bemenetet fogadjon a webes felhasználóktól, feldolgozza azt, majd HTML formátumú kimenetet állítson elő. A CGI-parancsfájlokat általában Perl nyelven írják, de a CGI más nyelvekkel, például a C-vel is képes együttműködni.

A webfejlesztésben a PHP is egyre népszerűbb nyelv. Ahogy a 20. órán majd látni fogjuk, az egyéni kiszolgálóoldali alkalmazások webes felületének ötlete egy egész programozási megközelítéshez, a webszolgáltatási környezetekhez vezetett. A webszolgáltatások programozásához számos vezető hardver- és szoftvergyártó – köztük a Sun, az IBM és a Microsoft – fejlesztett ki kifinomult rendszereket.

A Web fejlődésének másik fontos mérföldköve az XML (eXtensible Markup Language, bővíthető jelölőnyelv) megjelenése volt. Az XML nem korlátozódik előre meghatározott címkékre – a fejlesztő új címkéket is létrehozhat, amelyeknek a jelentését is ő határozhatja meg. Ez a rugalmasság a HTTP protokollt jelölőnyelvi szövegek átvitelére szolgáló eszközből általános, bármilyen típusú adat kézbesítésére alkalmas eszközzé változtatja. Az XML-ről a 20. fejezetben beszélünk bővebben.

Összefoglalás

Ezen az órán az Internet híres szolgáltatásának, a Világhálónak (World Wide Web) a működéséről tanultunk. Megismertük a HTML-dokumentumok és a HTTP protokoll felépítését, valamint bepillantást nyerhettünk a dinamikus HTML elvébe. A dinamikus HTML-ről és más webes eljárásokról többet is megtudunk majd a 20. és 21. fejezetekben.

Kérdezz-felelek

K *Milyen főbb részekből áll egy HTML-dokumentum?*

V A HTML-tartalmat a <HTML> és </HTML> címkék zárják közre. Ezek között a címkék között található a <HEAD> címkével jelzett fejrész, illetve a <BODY> címkével jelölt törzs. A <HEAD> rész adja meg a dokumentum címét, stílusait és egyéb vezérlési beállításait, míg a <BODY> a webböngésző ablakában megjelenő tartalmat foglalja magába. A szabvány egy !DOCTYPE utasítást is megkövetel az első HTML-elem előtt, de a dokumentumtípus-meghatározást gyakran elhagyják.

K *Melyik HTML-címke változtatja meg a szöveg színét?*

V Ha a szöveg színét szeretnénk módosítani, a <FONT> címkét kell használnunk a COLOR jellemzővel:

```
<FONT COLOR = "RED"> vörös szöveg </FONT>
```

K *Melyik HTML-címke határoz meg egy hiperhivatkozást?*

V A hiperhivatkozásokat az <A> címkével és annak HREF jellemzőjével adhatjuk meg:

```
<A HREF = "www.ElvisIsDiseased.com">I'm All Shook Up</A>
```

K *Miért van szükség a HTTP-ben egyeztetési szakaszra?*

V Ha a kiszolgáló és a böngésző más-más beállításokat alkalmaz a munkamenetekre, az egyeztetési szakasz során megegyezhetnek azokban a közös beállításokban, amelyek a sikeres kommunikációhoz szükségesek.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Böngésző** – HTTP-ügyfélprogram. A legtöbb ma használatos böngésző más protokollokat, például az FTP-t is képes értelmezni.
- **CGI (Common Gateway Interface, közös átjárófelület)** – Programozási felület, amely lehetővé teszi, hogy a fejlesztők parancsfájlokat és programokat ágyazzanak be a weboldalakba.
- **Címke** – HTML-utasítás.
- **Fej** – A HTML-dokumentum nyitó része, amelyben a dokumentum címe és más, nem kötelező paraméterek találhatók. A fejrészt a <HEAD> és </HEAD> címkék zárják közre.
- **Hiperhivatkozás** – Kiemelt rész egy weboldalon, amelyre kattintva a felhasználó a böngészőt egy másik dokumentumhoz vagy a hivatkozás meghatározásában URL-ként szereplő helyre irányíthatja.

- **HTML (Hypertext Markup Language, hiperszöveges jelölőnyelv)** – Jelölő- vagy leírónyelv, amelyen weboldalak készíthetők. A HTML szövegből és formázásra, hivatkozások meghatározására, illetve képek beágyazására szolgáló különleges kódokból áll.
- **HTTP (Hypertext Transfer Protocol, hiperszöveg-átviteli protokoll)** – A HTML-tartalomnak a kiszolgáló és az ügyfél között átvitelére szolgáló protokoll.
- **PHP** – A webfejlesztésben használt egyik népszerű programozási nyelv.
- **Törzs** – A HTML-dokumentumnak az a része, amelyik a böngészőablakban ténylegesen megjelenő szöveget tartalmazza. A törzsrészt a `<BODY>` és `</BODY>` címkék zárják közre.
- **URL (Uniform Resource Locator, egységes erőforráscím)** – Szabványos formátumú karakterlánc, amely egy erőforrást, illetve az annak eléréséhez használandó protokollt írja le. Az URL-eket a Világhálón található erőforrások azonosítására használják.

18. ÓRA

Elektronikus levelezés

A fejezet tartalmából:

- E-mail
- SMTP
- Levélszemét

Nem kell számítógépes szakembernek lennünk ahhoz, hogy észrevegyük, hogy a mai világban az elektronikus levelezés már a mindennapok elválaszthatatlan része. Mind a szakmai, mind a személyes kapcsolatok az e-mailekre támaszkodnak, mert nagy távolságra, gyors és megbízható kommunikációt biztosít. Ebben az órában az elektronikus levelezéssel kapcsolatos legfontosabb fogalmakat tekintjük át, valamint megmutatjuk, hogyan működnek az elektronikus levelezési szolgáltatások egy TCP/IP-hálózatban.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni egy elektronikus levél részeit.
- El tudjuk magyarázni az elektronikus levelek kézbesítésének folyamatát.
- Le tudjuk írni az SMTP-átvitel működését.
- El tudjuk magyarázni a POP3 és az IMAP4 levelezési protokollok működését.
- El tudjuk magyarázni, hogy mire valók a levelezőprogramok.

Mi az e-mail?

Az e-mail egy elektronikus levél, amit egy számítógépen megírnak, és egy hálózaton átküldenek egy másik számítógépre (ami lehet a közelben, de a világ másik felén is). Az elektronikus leveleket már a hálózatok történetének elején feltalálták. A komputermérnökök szinte abban a pillanatban, ahogy számítógépeket hálózatba kötöttek, már azon kezdtek töprengeni, hogy a számítógépek mellett lehetséges lenne-e az emberek közötti kommunikáció is ugyanezeken a hálózati kapcsolatokon keresztül.

A jelenlegi internetes levelezőrendszer története az ARPAnet idejébe nyúlik vissza. Az Internet elektronikus levelezőrendszere nagyrészt két, 1982-ben közzétett dokumentumon alapul: az RFC 821-en (Simple Mail Transfer Protocol, egyszerű levéltovábbítási protokoll) és az RFC 822-n (Standard for the Format of ARPA Internet Text Messages, szabvány az ARPA-hálózat szöveges üzeneteinek formátumához). A későbbi dokumentumok ezeket a szabványleírásokat finomították – ilyen volt például az RFC 2821, amely az SMTP új változatát határozta meg, valamint az RFC 2822 (Internet Message Format, internetes üzenetformátum). Az évek során más e-mail formátumokra is születtek javaslatok (ilyen volt például az X.400 rendszer, illetve több más, jogvédett formátum), de az egyszerűségének és a sokoldalúságának köszönhetően az SMTP alapú elektronikus levelezés vált uralkodóvá, majd az Internet *de facto* szabványává.

Az elektronikus levelezést a szöveg alapú felhasználói felületek korában találták fel, és az eredeti célja szövegek átvitele volt. Az e-mail üzenetformátumát tehát hatékony szövegátvitelre tervezték, és az elektronikus levelezés eredeti szabványai nem is fogalmaztak meg ajánlásokat a bináris fájlok küldésére. Az e-mail hatékonyságának egyik fő oka az, hogy az ASCII szöveg „könnyű" és egyszerűen továbbítható. Az ASCII szövegre fektetett hangsúly azonban végül korlátozónak bizonyult. Az 1990-es években az e-mail formátumát kibővítették, hogy bináris mellékleteket is lehessen használni. A levélmelléklet bármilyen típusú fájl lehet, feltéve, hogy a mérete nem haladja meg a levelezőprogram számára engedélyezett legnagyobb méretet. Ahogy ezen az órán megtanuljuk majd, a mellékleteket – amelyekben a felhasználók ma képfájlokat, táblázatokat, szövegszerkesztőben készített dokumentumokat és egyéb fájlokat csatolnak – általában MIME (Multipurpose Internet Mail Extensions, többcélú internetes levelezési bővítmények) formátumban kódolják.

Az elektronikus levelek formátuma

Az üzeneteinket a levelezőprogramunk alakítja az internetes átvitelhez szükséges formátumra. Az Interneten keresztül elküldött e-mailek két részből állnak: a fejlécből és a törzsből.

Az üzenet törzséhez hasonlóan a levélfejléc elküldése is ASCII szövegként történik. A fejléc kulcsszavas mezőnevek sorozatából áll, amelyeket egy vagy több, vesszővel elválasztott érték követ. A levélfejlécek legtöbb mezője már ismerős lehet azoknak, akik szoktak e-mailezni. A legfontosabb fejlécmezőket a 18.1. táblázatban soroltuk fel.

18.1. táblázat *A fontosabb levélfejléc-mezők*

Fejlécmező	Leírás
To: (Címzett)	A címzett(ek) e-mail címe(i).
From: (Feladó)	A feladó e-mail címe.
Date: (Dátum)	Az üzenet elküldésének dátuma és ideje.
Subject: (Tárgy)	Az üzenet tárgyának rövid leírása.
Cc: (Másolatot kap)	Az üzenetből másolatot kapó egyéb felhasználók e-mail címei.
Bcc: (Titkos másolatot kap)	Azoknak a felhasználóknak az e-mail címei, akik titkos másolatot kapnak az üzenetből. A titkos másolat a levél olyan másolata, amelyről a többi címzett nem tud. A Bcc mezőben felsorolt e-mail címek nem jelennek meg a többi címzett által kapott fejlécekben.
Reply-To: (Válaszcím)	Az az e-mail cím, ahová az üzenetre adott választ küldeni kell. Ha ezt a mezőt nem adjuk meg, a válaszok a From: mezőben szereplő címre érkeznek.

A fejlécet egy üres sor követi, majd az üzenet törzse (vagyis az elektronikus levél szövege) következik. A felhasználók azonban gyakran nem csupán szöveget akarnak küldeni az e-mailben. A bináris fájlok e-mailben történő továbbítására számos módszer született. A régebbiek a bináris biteket az ASCII-megfelelőjükre alakítják át: az eredményként kapott fájl úgy néz ki, mint egy ASCII szöveg – valójában az is –, de nem tudjuk elolvasni, mert az eredeti bináris kódot jelképező betűk számunkra értelmetlen halmazából áll. Ezt a módszert az eredetileg a Macintosh-hoz kifejlesztett BinHex, valamint az Unixhoz készített Uuencode segédprogram használja. Ahhoz, hogy a fájlt visszaalakíthassuk az eredeti bináris formájára, nekünk vagy a levelezőprogramunknak rendelkeznünk kell a megfelelő visszafejtő segédprogrammal.

A bináris fájlok e-mailben történő átvitelére általánosabb és szélesebb körben használható megoldást jelent a MIME formátum. A MIME olyan általános formátum, amely az internetes elektronikus levelek képességeit bővíti. A MIME kódolást értelmezni

képes levelezőprogramok a bináris mellékleteket átvitel előtt MIME formátumban kódolják, amikor pedig az üzenet megérkezik a címzetthez, a címzett számítógépén található MIME-képes levelezőprogram visszafejti a kódolást, és a mellékletet visszaalakítja az eredeti formájára.

A MIME többek között az alábbi újdonságokat nyújtja az internetes levelek számára:

- Bővített karakterkészletek. A MIME nem korlátozódik a szabványos, 128 karakteres ASCII készletre. Ez azt jelenti, hogy különleges, illetve az amerikai angol nyelvben nem szereplő karakterek átvitelére is használhatjuk.
- Korlátlan hosszúságú sorok és üzenet.
- Szabványos kódolás a mellékletek számára.
- Lehetőség képek, hangok, hivatkozások és formázott szöveg beágyazására az üzenetbe.

A legtöbb elektronikus levelezőprogram támogatja a MIME kódolást. A MIME formátum leírását több RFC tartalmazza.

Az elektronikus levelezés működése

Más internetes szolgáltatásokhoz hasonlóan az elektronikus levelezés is ügyfél-kiszolgáló alapú folyamat, az e-mailek működése azonban egy kicsit bonyolultabb. Röviden összefoglalva: a levéltranzakciók két végén elhelyezkedő számítógépek egyaránt ügyfélként működnek, az üzenetet pedig a közöttük álló kiszolgálók továbbítják a hálózaton.

Az elektronikus levelek kézbesítésének folyamatát a 18.1. ábrán láthatjuk. Egy ügyfél üzenetet küld egy levélkiszolgálónak. A kiszolgáló elolvassa a címzett címét, és továbbítja az üzenetet a címhez társított másik levélkiszolgálónak. Az üzenet a célkiszolgálón egy postafiókban (mailbox) tárolódik. (A *postafiók* a beérkező levelek mappája vagy várakozási sora.) A felhasználó, akinek az üzenet szól, időnként bejelentkezik a levélkiszolgálóra, hogy megnézze, jöttek-e levelei. Régebben az volt a szabványos eljárás, hogy a felhasználó számítógépén található ügyfélprogram letöltötte a felhasználó postafiókjában várakozó üzeneteket, amelyeket aztán a felhasználó elolvashatott, elraktározhatott, törölhetett vagy továbbíthatott, illetve válaszolhatott rájuk. Bár ez a megoldás még ma is elterjedt, az olyan újabb megoldások, mint az IMAP és a webmail, lehetővé teszik, hogy a felhasználók a kiszolgálón kezeljék a leveleiket, anélkül, hogy valaha is le kellene tölteniük azokat.

Ahogy az óra későbbi részében látni fogjuk, az ügyfélalkalmazás, amelyet *levelezőprogramnak* (vagy levélolvasónak, email reader) hívnak, gondoskodik a kimenő levelek elküldéséről, illetve a bejövő levelek letöltéséhez szükséges bejelentkezésről a kiszol-

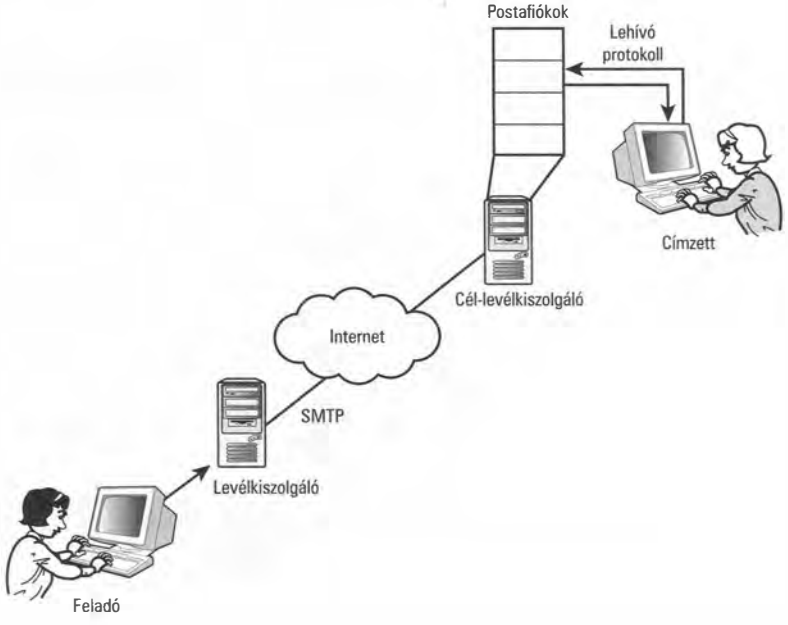
gálóra. A legtöbb felhasználó egy levelezőprogram felületén keresztül bonyolítja a levelezését. Az üzenetek elküldését és a kiszolgálók közötti továbbítását az SMTP (Simple Mail Transfer Protocol) levéltovábbítási protokoll kezeli.

Az üzenet továbbításához szükséges információkat az e-mail cím adja meg a kiszolgálónak. A népszerű internetes e-mail címek formátuma a következő:

```
felhasználó@kiszolgáló
```

Vagy (például):

```
BillyBob@Klondike.net
SallyH@montecello.com
cravenprof@harvard.edu
```



18.1. ábra

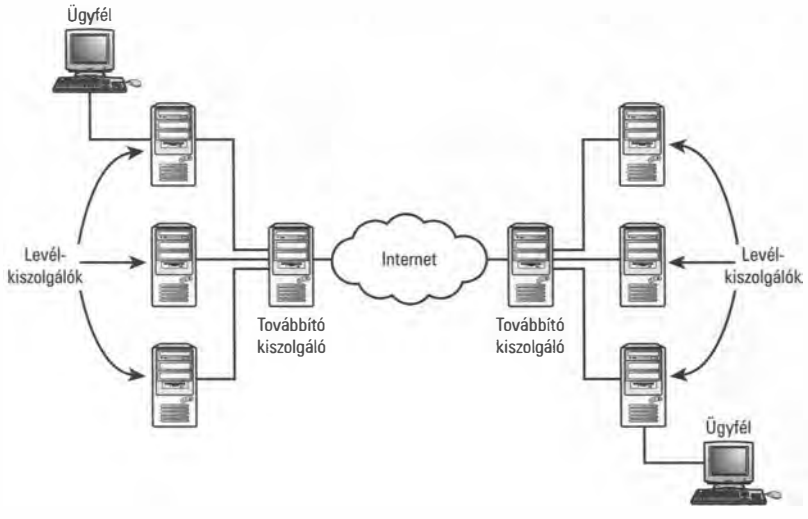
Az elektronikus levelek kézbesítésének folyamata

Ebben a szabványos formátumban a kukacjel (@) után álló szöveg a cél-levélkiszolgáló neve, a kukac előtti szöveg pedig a címzett postafiókjának neve a levélkiszolgálón.

A kukacjel utáni szöveg valójában többnyire a címzett tartományában alapértelmezett levélkiszolgáló tartománynevét jelöli. A tartomány névkiszolgálói (DNS-kiszolgálói) egy MX-erőforrásrekordot tárolnak, amely a tartománynévhez egy levélkiszolgálót társít. A DNS-ről a 11. fejezetben beszéltünk bővebben.

Az e-mail cím formátuma rávilágít egy fontos dologra, amit tudnunk kell az internetes levelezésről: az elektronikus levelek célja nem a címzett számítógépe, hanem a címzett postafiókja a levélkiszolgálón. Az utolsó lépés, amelynek során a várakozó levelek a levélkiszolgálóról a címzett számítógépére kerülnek, valójában külön folyamat. Az óra későbbi részében megtanuljuk majd, hogy ezt az utolsó lépést egy olyan levéllehívó protokollon keresztül bonyolítjuk, mint a POP (Post Office Protocol, postahivatal-protokoll) vagy az IMAP (Internet Message Access Protocol, internetes üzenethozzáférési protokoll).

Egyes hálózatok több, alá-fölérendelt viszonyban álló levélkiszolgálóval biztosítják a hatékonyabb kézbesítést. Ebben a forgatókönyvben (lásd a 18.2. ábrát) a helyi levélkiszolgáló egy továbbító kiszolgálónak (relay server) küldi el az üzenetet, az pedig egy másik, a célhálózaton található továbbító kiszolgálónak adja át azt, és ez a továbbító kiszolgáló juttatja el a levelet a címzetthez társított helyi kiszolgálóhoz.



18.2. ábra

A továbbító kiszolgálók gyakran hatékonyabbá teszik a levelek kézbesítésének folyamatát

SMTP (Simple Mail Transfer Protocol)

Az SMTP az a protokoll, amelyet a levélkiszolgálók a leveleknek egy TCP/IP-hálózaton való továbbítására használnak. A levelet küldő ügyfélszámítógép szintén az SMTP segítségével adja át a levelet egy helyi kiszolgálónak kézbesítésre.

A felhasználónak soha nem kell megtanulnia az SMTP nyelvén beszélni, mert az SMTP-kommunikáció a színfalak mögött zajlik. Mindazonáltal néha nem árt, ha tudunk eztazt az SMTP-ről, hogy képesek legyünk értelmezni a kézbesítetlen levelekhez társuló

hibaüzeneteket. Ezenkívül egyes programok és parancsfájlok időnként közvetlenül érik el az SMTP-t, hogy figyelmeztetéseket és riasztásokat küldjenek a hálózat karbantartó személyzetének.

Más TCP/IP-alkalmazásszolgáltatásokhoz hasonlóan az SMTP is a TCP/IP-protokollvermen keresztül társalog a hálózattal. A levelezőprogramok feladatai egyszerűek, mert a TCP/IP-protokollszoftver kapcsolati és ellenőrzési szolgáltatásaira támaszkodhatnak. Az SMTP-kommunikáció egy TCP-kapcsolaton keresztül zajlik, amely az SMTP-kiszolgáló 25-ös kapujára irányul. Az ügyfél és a kiszolgáló közötti párbeszéd az ügyféltől érkező szabványos, négykarakteres parancsokból (és adatokból), valamint a kiszolgálótól kapott három számjegyű válaszkódokból áll. A fontosabb SMTP-ügyfélparancsokat a 18.2. táblázatban láthatjuk, a megfelelő kiszolgálói válaszkódokat pedig a 18.3. táblázatban soroltuk fel.

18.2. táblázat *Az SMTP ügyfélparancsai*

Parancs	Leírás
`HELO`	Köszöntés. (Az ügyfél kapcsolatot kezdeményez a kiszolgálóval.)
`MAIL FROM:`	A feladó felhasználó e-mail címét előzi meg.
`RCPT TO:`	A fogadó felhasználó e-mail címét előzi meg.
`DATA`	Az üzenettartalom átvitelének megkezdésére irányuló szándékot jelenti be.
`NOOP`	A kiszolgálót OK válasz küldésére kéri.
`QUIT`	A kiszolgálót OK válasz küldésére és a munkamenet bezárására kéri.
`RESET`	Megszakítja a levélküldést.

18.3. táblázat *Néhány SMTP-kiszolgálói válaszkód*

Kód	Leírás
`220`	A tartományi szolgáltatás készen áll.
`221`	A tartományi szolgáltatás bezárja az átviteli csatornát.
`250`	A kért művelet sikeresen befejeződött.
`251`	A felhasználó nem helyi. Az üzenet a `<path>`-ban (elérési út) megadott címre lesz továbbítva.
`354`	Az adatküldés megkezdhető. Az adatok végét a `<CRLF>.<CRLF>` karakterlánccal kell jelezni (ez egy külön sorban levő pontot jelent).
`450`	Nem került sor művelet végrehajtására, mert a postafiók elfoglalt.
`500`	Nyelvtani hiba: a parancs nem ismerhető fel.
`501`	Nyelvtani hiba: gond van a paraméterekkel vagy argumentumokkal.
`550`	Nem került sor művelet végrehajtására, mert a postafiók nem található.
`551`	A felhasználó nem helyi. Az üzenetet próbálja ide küldeni: `<path>`.
`554`	A tranzakció meghiúsult.

Az alábbiakban nagy vonalakban ismertetjük azt a folyamatot, amelynek során egy levél a levélkiszolgálóhoz kerül. Ahogy az óra korábbi részében említettük, ez az eljárás nem csak arra szolgál, hogy a kezdeményező ügyfél elküldjön egy levelet a helyi levélkiszolgálónak, hanem arra is, hogy a helyi kiszolgáló továbbítsa a levelet a célkiszolgálónak vagy egy a továbbítási útvonalon található másik kiszolgálónak:

1. A küldő számítógép egy `HELO` parancsot küld a kiszolgálónak, argumentumként átadva a küldő nevét.
2. A kiszolgáló visszaadja a `250` válaszkódot.
3. A küldő kiadja a `MAIL FROM:` parancsot, argumentumként az üzenetet küldő felhasználó e-mail címét átadva.
4. A kiszolgáló visszaadja a `250` válaszkódot.
5. A küldő kiadja az `RCPT TO:` parancsot, argumentumként az üzenet címzettjének e-mail címét átadva.
6. Ha a kiszolgáló képes leveleket fogadni a címzett számára, a `250` válaszkódot adja vissza, egyébként pedig egy a probléma okát leíró kódot (az `550` például azt jelenti, hogy a felhasználó postafiókja nem található).
7. A küldő kiadja a `DATA` parancsot, jelezve, hogy készen áll a levéltartalom elküldésének megkezdésére.
8. A kiszolgáló visszaadja a `354` válaszkódot, amivel arra utasítja a küldőt, hogy kezdje meg az üzenettartalom elküldését.
9. A küldő elküldi a levél tartalmát, amelyet egy külön sorban levő ponttal (.) zár le.
10. A kiszolgáló visszaadja a `250` válaszkódot, jelezve, hogy megkapta a levelet.
11. A küldő kiadja a `QUIT` parancsot, ami azt jelzi, hogy az átvitel befejeződött, és a munkamenetet be kell zárni.
12. A kiszolgáló visszaadja a `221` válaszkódot, jelezve, hogy be fogja zárni az átviteli csatornát.

A hálózat ezt az SMTP-kommunikációs eljárást használja arra, hogy az elektronikus levelet eljuttassa a felhasználó postafiókjába a cél-levélkiszolgálón. A levél ezután az említett postafiókban várakozik, amíg a felhasználó be nem jelentkezik, hogy megtekintse a postáját. A levelezőprogram típusától, illetve az általa használt lehívó protokolltól függően a felhasználó vagy letölti a levelet megtekintésre és feldolgozásra a számítógépére, vagy közvetlenül a kiszolgálón kezeli és szerkeszti azt.

A levelek lehívása

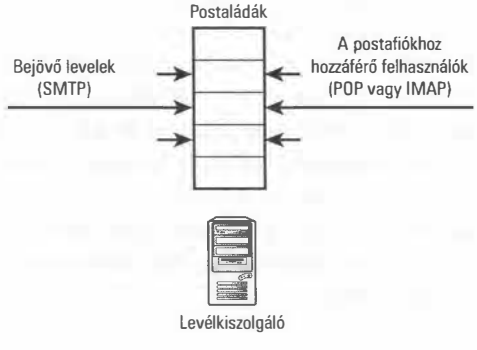
Az SMTP-kézbesítésnek az előző részben ismertetett folyamatát nem arra tervezték, hogy felhasználóknak kézbesítsen üzeneteket, hanem arra, hogy a leveleket a felhasználók postafiókjaiba juttassa. Ahhoz tehát, hogy megtekinthesse a leveleit, a felhasználónak hozzá kell férnie a postafiókjához. Ez az újabb lépés kicsit bonyolultabbá teszi az eljárást, viszont a következő előnyökkel jár:

- A kiszolgáló akkor is fogadhat további üzeneteket a felhasználó számára, amikor a felhasználó számítógépe nem kapcsolódik a hálózatra.
- A levélkézbesítő rendszer független a címzett számítógépétől vagy helyétől.

Ez utóbbi előnnyel az e-mailezők többsége tisztában van, hiszen ez teszi lehetővé, hogy a felhasználó több helyről is hozzáférhessen a leveleihez. Elméletben bármely internetkapcsolattal és levelezőprogrammal rendelkező számítógépet be lehet állítani úgy, hogy ellenőrizze a felhasználó postafiókjába érkező postát: otthonról, a munkahelyünkről vagy akár egy szállodai szobából is megnézhetjük, hogy érkezett-e levelünk. A postafiók eléréséhez és a levelek letöltéséhez azonban egy levéllehívó protokollra van szükség – a következőkben két ilyen protokollal, a POP (Post Office Protocol) és az IMAP (Internet Message Access Protocol) nevűekkel ismerkedünk meg, valamint egy újabb lehetőséggel, a webmaillel, amely a postafiók elérését egy közönséges webböngészőn keresztül teszi lehetővé.

A valóságban az olyan hálózati biztonsági megoldások, mint a tűzfalak, néha megakadályozzák, hogy a felhasználó ismeretlen helyről is hozzáférjen a postájához, vagy leveleket küldjön.

A felhasználói postafiókokat tároló levélkiszolgálónak általában mind az SMTP szolgáltatást, mind egy levéllehívó protokollt támogatnia kell – az előbbit a beérkező üzenetek fogadásához, míg az utóbbit a postafiók eléréséhez. A folyamatot a 18.3. ábra mutatja. Az eljáráshoz az SMTP és a levéllehívó szolgáltatás összeegyeztethetősége és összehangolása szükséges, hogy az adatok ne vesszenek el, vagy ne sérüljenek, ha ezek a szolgáltatások egyidejűleg férnének hozzá ugyanahhoz a postafiókhoz.



18.3. ábra
Az SMTP-kiszolgálóprogramnak és a levéllehívó programnak össze kell hangolnia a postafiókhoz való hozzáférést

POP3

A POP3 (Post Office Protocol version 3) széles körben használt levéllehívó protokoll. A POP3 leírása az RFC 1939-ben található, amelyet későbbi RFC-dokumentumok bővítettek és finomítottak. A működése a következő: az ügyfél TCP-kapcsolatot kezdeményez a levélkiszolgálón található POP3-kiszolgálói alkalmazással, a POP3-kiszolgáló

pedig a 110-es TCP-kapun figyeli a kapcsolatokat; miután a kapcsolat létrejött, az ügyfélprogramnak el kell küldenie a felhasználónevet és a jelszót a levélkiszolgálónak; ha a bejelentkezési adatokat a kiszolgáló elfogadja, a felhasználó hozzáférhet a postafiókjához, hogy leveleket töltsön le vagy töröljön.

Az SMTP-ügyfélhez hasonlóan a POP3-ügyfél is négykarakteres parancsokat használ a kiszolgálóval való kommunikációhoz. A kiszolgáló néhány rövid szóval válaszol – a `+OK` például azt jelzi, hogy a kiszolgáló végrehajtotta a parancsot, míg az `-ERR` azt, hogy a parancs hibát eredményezett. A válaszok ezenkívül további argumentumokat vagy paramétereket is tartalmazhatnak. A postafiókban minden üzenetre egy számmal kell hivatkozni. Az ügyfél a `RETR` (retrieve, lehívás) paranccsal tölthet le egy levelet a kiszolgálóról, míg a `DELE` (delete, törlés) parancs töröl egy üzenetet onnan.

A POP3-ügyfél és a kiszolgáló között kicserélt üzenetek a felhasználó számára láthatatlanok; ezeket a parancsokat a levelezőprogram a felhasználónak a program felületén végrehajtott műveleteire válaszolva adja ki.

A POP3 egyik hátránya, hogy korlátozott számú műveletet tud végrehajtani a kiszolgálón. A felhasználó csak annyit tehet, hogy kiíratja a postafiókban tárolt üzeneteket, leveleket töröl, vagy leveleket tölt le. Az üzenetek tartalmát szerkeszteni azonban csak az ügyféloldalon lehet. Ez a korlátozás késlekedést és megnövekedett hálózati forgalmat okozhat, mivel a leveleket le kell tölteni a kiszolgálóról az ügyfélre. Az újabb és kifinomultabb IMAP protokollt azért fejlesztették ki, hogy kiküszöbölje ezen hiányosságok némelyikét.

IMAP4

Az IMAP4 (Internet Message Access Protocol version 4) a POP3-hoz hasonló levéllehívó protokoll, amely azonban több olyan új szolgáltatást is nyújt, ami a POP3-ban nem érhető el. Az IMAP4 segítségével tallózhatunk a kiszolgálón levő mappákban, és a leveleket anélkül helyezhetjük át, törölhetjük és tekinthetjük meg, hogy előbb le kellene töltenünk azokat a saját számítógépünkre. Az IMAP4 ezen kívül bizonyos beállításoknak, például az ügyfélablak megjelenésének vagy a kiszolgálón végrehajtott keresésekhez használt keresőkifejezéseknek a mentését is lehetővé teszi, és a postafiókok létrehozására, törlésére és átnevezésére is lehetőségünk van a kiszolgálógépen.

Az újabb levelezőprogramok többsége egyaránt ismeri a POP3-at és az IMAP4-et. Bár a POP3-at jelenleg többen használják, az IMAP számos előnye garantálja, hogy egyre több program fog átállni az IMAP4 protokollra.

Levelezőprogramok

A levelezőprogram egy ügyfélalkalmazás, amely a felhasználó munkaállomásán fut, és egy levélkiszolgálóval kommunikál. Ahogy az óra korábbi részében megtanultuk, a helyi munkaállomás nem kerül közvetlen kapcsolatba a levelek címzettjeivel, hanem a levélkiszolgálónak küldi el az üzeneteket a levelezőprogram segítségével, és a kiszolgáló juttatja el azokat a címzetthez rendelt levélkiszolgálóhoz. Szokványos levélküldésnél a levél címzettje bejelentkezik majd a levélkiszolgálón található személyes postafiókjába, és letölti a leveleket a saját munkaállomására. Az eljárás első és utolsó lépését (az üzenet elküldését az eredeti kiszolgálónak, illetve a levél letöltését a fogadó kiszolgálóról) jellemzően egy levelezőprogram bonyolítja.

A levelezőprogramnak három feladata van:

- Elküldi a kimenő leveleket egy levélkiszolgálónak az SMTP segítségével.
- Lehívja a beérkező leveleket egy levélkiszolgálóról a POP3 vagy az IMAP használatával.
- Felhasználói felületként szolgál a levelek olvasásához, írásához és kezeléséhez.

A levelezőprogramnak képesnek kell lennie mind SMTP-, mind levéllehívó (POP- vagy IMAP-) ügyfélként működni.

Az órában korábban ismertetett levelezési protokollok világos térképet adnak az elektronikus levelezéshez, ezért minden levelezőprogram ugyanúgy működik. A programok beállításának módja különbözhet, de ha tisztában vagyunk a fejezetben ismertetett eljárásokkal, akkor általában nem nehéz kitalálni, hogyan működnek. Más hálózati ügyfélprogramokhoz hasonlóan a levelezőprogramok is a protokollvermen keresztül kommunikálnak a hálózattal. A számítógépnek tehát, amelyen a levelezőprogram fut, működő TCP/IP-megvalósítással kell rendelkeznie, és úgy kell beállítani, hogy a levelezőprogram a TCP/IP-n keresztül elérhesse a hálózatot.

Miután meggyőződtünk róla, hogy a számítógépünk megfelelően működik ügyfélként egy TCP/IP-hálózaton, meg kell tudnunk néhány további paramétert a hálózat valamelyik illetékesétől, hogy beállíthassuk a levelezőprogramot a rendszerünkön. Ha otthoni felhasználók vagyunk, ezeket az információkat az internetszolgáltatónktól szerezhetjük be, ha pedig céges számítógépet használunk, akkor a hálózat rendszergazdájától.

A következőket kell megtudnunk:

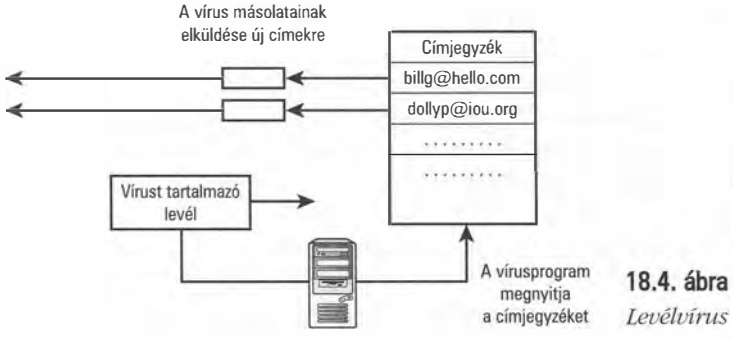
- A kimenő levelekhez használt levélkiszolgáló teljesen minősített tartománynevét. Ennek a kiszolgálónak a gépneve (állomásneve) általában `SMTP`, amelyet a tartomány neve követ (például: `SMTP.rosbud.org`).

- A POP- vagy IMAP-kiszolgáló teljesen minősített gépnevét.
- A POP- vagy IMAP-kiszolgálón levő levelezési fiókunkhoz tartozó felhasználónevet és jelszót.

A levelezőprogram beállítása lényegében ezeknek az információknak a megszerzéséből és a programba való beírásából áll.

A levelezőprogramokat a legtöbb operációs rendszeren fokozatosan beépítették a szabványos asztali környezetbe. A Windows-felhasználók a leveleket a Windows Mail vagy az Outlook ügyfélprogramon keresztül érhetik el, a Mac OS X rendszereken az Apple Mail a szabványos alkalmazás, a Linux rendszerek pedig általában egy olyan népszerű, nyílt forrású ügyfélprogramot tartalmaznak, mint az Evolution vagy a Mozilla Thunderbird.

A levelezőprogramokat gyakran építik egybe hasonló eszközökkel, amelyek naptárat, ütemezési lehetőségeket vagy címjegyzék-szolgáltatásokat nyújtanak. A levelezőprogramok emellett a fájlnév-kiterjesztéseket (`.doc`, `.txt`, `.pdf`, `.jpg`) is képesek értelmezni, így el tudják indítani a megfelelő megjelenítőprogramot a beérkező mellékletek megtekintéséhez. Ez a fajta, más alkalmazásokkal való egybeépítés kényelmes, ha megfelelően használjuk, de egyben a ***makróvírusok*** egy teljesen új nemzedékét hívta életre: ezek a vírusok elsősorban a Windows rendszereket fenyegetik, és a levelek mellékleteiben érkeznek. A makróvírusok általában új e-mail címeket olvasnak ki a felhasználó címjegyzékéből, majd önműködően elküldik magukat az ott szereplő címekre (lásd a 18.4. ábrát).



18.4. ábra
Levélvírus

A vírusoknak ez a típusa jelentős károkat okozott a múltban, de az utóbbi években az óvatosabbá váló felhasználóknak és a hatékony víruskereső megoldásoknak köszönhetően a probléma kezelhetőbbé vált. Fontos, hogy megjegyezzük, hogy a mellékletek fogadása és a levélben kapott hivatkozásokra történő kattintás kockázatot jelent a rendszerünkre nézve. Olvassuk el az operációs rendszerünk gyártójának ajánlásait, hogy megtudjuk, hogyan kell beállítanunk a rendszerünket ahhoz, hogy ezt a kockázatot a lehető legkisebbre csökkentsük.

Webmail

A Világháló felemelkedése az elektronikus levelezés egy teljesen új fajtáját hívta életre, amely a webes technológia köré épül. A Web alapú (webmail) levelezőeszközök nem igényelnek levélolvasó ügyfélprogramot: a felhasználó egyszerűen ellátogat a levelezési webhelyre egy internetböngészővel, és a postáját egy webes felületen keresztül érheti el. A felhasználó tehát bármely internetkapcsolattal rendelkező számítógépről hozzáférhet a leveleihez. A Hotmail, a Yahoo Mail vagy a Google Gmail alkalmazása ilyen webmail-szolgáltatások. Ezek a szolgáltatások sokszor ingyenesek – vagy majdnem ingyenesek –, mert a szolgáltató elég pénzt keres a reklámokon ahhoz, hogy támogassa a rendszert.

A webmail sokoldalú és könnyen használható. Jó választás azoknak a laikus otthoni felhasználóknak, akik gyakran szörfölnek a Weben, és nem szeretnének levelezőprogramok beállításával és hibaelhárításával vesződni. Egyes cégek bizonyos helyzetekben ma már webes levelezőprogramot használnak, mert a tűzfaluk átengedi a HTTP-forgalmat, de letiltja az SMTP-t. A webmail első pillantásra nem biztonságosnak tűnhet, hiszen az Interneten mindenki tudja, hogyan érheti el a Yahoo! webhelyét, így aztán valószínűleg azt is ki tudja találni, hogyan férhet hozzá a Yahoo! postaszolgáltatásához. Fontos azonban látnunk, hogy a hagyományos elektronikus levelezés sem igazán biztonságos, hacsak nem teszünk lépéseket a biztonságossá tételére. Bárki, aki ismeri a felhasználónevünket és a jelszavunkat, megnézheti a leveleinket. A nagyobb webmail-szolgáltató webhelyek biztonságos bejelentkezést és egyéb biztonsági intézkedéseket nyújtanak. Ha egy kis, helyi webmail-szolgáltatás igénybe vételén gondolkodunk, nem árt, ha utánanézünk, mennyire biztonságos a rendszerük.

A webmaillel kapcsolatban a legtöbb panasz a szolgáltatás sebességére érkezik. Mivel ez a levelezőrendszer igazából „nincs jelen" az ügyfélszámítógépen (a webböngészőt kivéve), a levelek megírása, megnyitása és mozgatása egyaránt a hálózati kapcsolat szűk keresztmetszetén át történik. Ezzel szemben a hagyományos levelezőprogramok a munkamenet kezdetén letöltik az új leveleket, és a levélírással és a levelek tárolásával kapcsolatos minden műveletet az ügyfélen hajtanak végre. Ha viszont eltekintünk a lassabb működéstől, a webmail rendkívül kényelmes, ami biztosítja, hogy a webmail előnyös maradjon sok internetfelhasználó számára.

A webmail elsődleges feladata természetesen az, hogy lehetővé tegye a felhasználónak, hogy üzeneteket küldjön és fogadjon. Bár a webmail elve teljesen újnak tetszhet, valójában nem sokban különbözik a közönséges e-mail rendszerétől, amelyet a 18.1. ábrán ábrázoltunk. A különbség csupán annyi, hogy a webmail esetében a levelek küldésére és fogadására szolgáló szoftver a levélkiszolgálón működik, és a címzett ezt a szoftvert egy webes felületen keresztül érheti el. A színfalak mögött azonban a webmail-rendszerek is az SMTP-t használják a levelek elküldésére a hálózaton át.

Levélszemét

Az elektronikus levelezés fejlődésére semmi sem volt akkora hatással, mint a levélszemét (spam) megjelenése. A *levélszemét* kifejezés azokat a nagy tömegben elküldött leveleket takarja, amelyek internetfelhasználók millióinak tömik el a postafiókját. Ezek az üzenetek bankkölcsönöket, diétákat, különféle termékeket és szolgáltatásokat reklámoznak, ál-jótékonysági felhívásokat tesznek közzé, és az örök üdvösséget ígérik. Technikai szempontból a levélszemét is csak elektronikus posta – ezért is működik. A leveleket továbbító levélkiszolgálók nem tudhatják, hogy egy üzenetet egy rosszindulatú automatikus program írt, vagy a címzett kedvese.

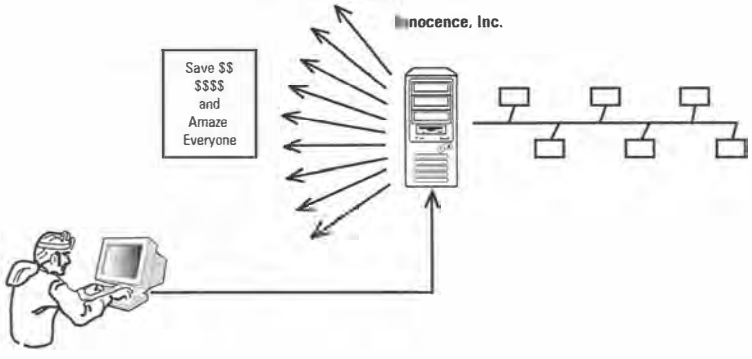
Szerencsére azonban a címzettnek több lehetősége is van arra, hogy azonosítsa a levélszemetet, és megszabaduljon tőle. A levélszemét elleni küzdelemben használt módszerek között vannak olyanok is, amelyek a TCP/IP elveire épülnek, így könyvünk témájába vágnak. A levélszemét-küldözgetők ugyanakkor – amint látni fogjuk – leleményesek abban, hogy megtalálják a kiskapukat a védelmünkön, ezért egyetlen megoldás sem tart örökké. Az újabb módszerek elsősorban a levelek szövegének elemzésére támaszkodnak.

A levélszemét-üzlet beindulásának idején a címzettek hamarosan rájöttek, hogy a levélszemét jelentős része néhány konkrét e-mail címről érkezik. A levélszemét-elhárítással foglalkozók hatalmas címadatbázisokat állítottak össze, amelyekben összegyűjtötték azokat a címeket, amelyeket a levélszeméthez lehetett kapcsolni. A tűzfalak, levélkiszolgálók és ügyfélprogramok a beérkező leveleket ezeknek a feketelistára tett címeknek az alapján ellenőrizhették.

A levélszemetelők azonban gyakran változtatják az IP-címüket és a tartománynevüket, hogy elkerüljék a feketelistára kerülést. A feketelista a védelem első vonalaként megfelel, de a levélszemét teljes kiszűrésére nem elegendő. Valójában a hagyományos feketelisták egyre kevésbé számítanak, mert a levélszemetelők tökéletesítették a megkerülésükre szolgáló módszereket. Az egyik ilyen módszer az, hogy gyanútlan cégek levélkiszolgálóit használják a levélszemét továbbítására. Ahogy az óra korábbi részében megtanultuk, az SMTP-levélkiszolgálók egyszerűen várakoznak az ügyfelek leveleire, és továbbítják azokat. Elvben természetesen csak a tulajdonosa használhatná a kiszolgálót üzenetek továbbítására, de egy nem tökéletesen elzárt levélkiszolgálót *bárki* használhat – beleértve a máshonnan támadó levélszemét-küldőket is (lásd a 18.5. ábrát). Néha teljesen ártatlan cégek és magánszemélyek is egy feketelistán találják magukat, mert a levélszemetelők az ő kiszolgálójukat használják közvetítőként.

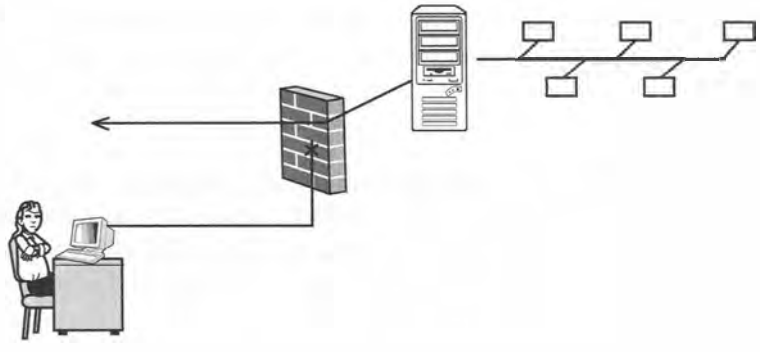
A levélszemét ellen küzdők ez ellen a taktika ellen a következő megoldást eszelték ki: a levélkiszolgálót a vállalati tűzfal mögött helyezik el, és a bejövő SMTP-kérelmeket megállítják a tűzfalnál (lásd a 18.6. ábrát), így a cég megvédheti magát attól, hogy levélszemét-továbbítóvá váljon. Ahogy a 18.6. ábrán láthatjuk, a tűzfalon belül levő

ügyfélprogramok továbbíthatnak üzeneteket a levélkiszolgáló segítségével, a tűzfalon kívülről azonban a levélkiszolgálót nem érhetik el. Ez a megoldás alkalmas a levélszemét kiszűrésére, de van néhány hátránya. Ha egy otthoni felhasználó utazás közben, hordozható számítógépről próbál leveleket küldeni, előfordulhat, hogy nem jár sikerrel, hacsak nem állítja át a levelezőprogramját, hogy egy másik SMTP-kiszolgálóra mutasson.



18.5. ábra

A levélszemét-küldözgetők néha valaki más nem biztosított és gyanútlan levélkiszolgálóját használják az üzeneteik elküldésére



18.6. ábra

Ha az SMTP-kiszolgálót egy tűzfal mögött helyezzük el, és blokkoljuk a bejövő SMTP-kérelmeket, megvédhetjük a kiszolgálót a levélszemét-küldők visszaéléseitől

Egyes levélszemetelők egyenesen betörnek az ártatlan felhasználók számítógépére, és úgy állítják át a rendszert, hogy levélszemetet küldjön. Ezek a levélszemét-robotok (spambot) gyakran levelek ezreit küldik el, mire felfedezik őket. A hálózati rendszergazdák a *fehérlisták* alkalmazásával vágtak vissza – ezek azokat a címeket tartalmazzák, amelyek számára *megengedett*, hogy levelet küldjenek a tartománynak. Ez a módszer hatékony lehet, de sok cég számára túlságosan korlátozó.

Egy másik védelmi módszert jelentenek a *szürkelisták*. A szürkelistás rendszerek átmenetileg visszautasítják az ismeretlen forrásból érkező üzeneteket. Ha az üzenet érvényes, a feladó kiszolgáló újraküldi azt – a levélszemét-küldő kiszolgálók azonban jellemzően automatizált eszközök, amelyeket nem arra terveztek, hogy meghiúsult kézbesítés esetén újra elküldjék az üzenetet. Ha tehát a kiszolgáló nem küldi el újra az üzenetet, feltételezhetjük, hogy levélszemétről van szó. Mire egy levélszemét-kiszolgáló mégis eljutna az újraküldésig, jó esély van rá, hogy az Internet feketelistázó szolgáltatásai rögzítik a címét. A szürkelistákat ezért gyakran a feketelistákkal együtt használják.

A levélszemét elleni küzdelemben sok eszköz az üzenettartalom elemzésére támaszkodik, az ilyen levelek fejlécében és üzenettörzsében ugyanis bizonyos kifejezések sűrűbben fordulnak elő. Egyes levélszemét-szűrők szabályok alapján tiltják le az üzeneteket. Egy szűrő kizárhatja például a szitokszókat vagy azokat az egyéb kifejezéseket, amelyek indokolatlanul használt anatómiai fogalmakhoz kapcsolódnak. Más, kifinomultabb eljárások, például a Bayes-féle levélszemét-szűrés, valószínűségszámításon alapuló módszerekkel elemzik a levelek szóhasználatát, és egy pontszámmal adják meg, hogy a levél mekkora valószínűséggel levélszemét. Egyes kéretlen levelek furcsa szóhasználata és rejtélyes nyelvezete arra utal, hogy a küldő szeretne átcsúszni ezeknek a tartalom alapú valószínűségi szűrőknek a hálóján.

Egyes szűrőeszközök hajlamosak hamis pozitív eredményeket adni, és érvényes üzeneteket kizárni, ha azok a kéretlen levelekhez hasonló profilt mutatnak. A legjobb eszközök lehetőséget adnak a szűrő „betanítására" a hibás pozitív eredmények megmutatásával, így a szűrő újraszámíthatja a valószínűségeket, és nem követi el kétszer ugyanazt a hibát.

Összefoglalás

Ebben az órában azt tárgyaltuk, hogy mi történik az elektronikus levelekkel, miután elhagyták a számítógépünket, és a színfalak mögé pillantva megismertük az e-mailek kézbesítésének folyamatát. Tanultunk az SMTP-ről, valamint az olyan levéllehívó megoldásokról, mint a POP3, az IMAP4 vagy a webmail. Az órán ezen kívül tisztáztuk a levelezőprogramok szerepét, és beszéltünk a levélszemét elleni küzdelemben alkalmazott módszerekről.

Kérdezz-felelek

K *Képes vagyok leveleket küldeni, de nem tudok kapcsolódni a levélkiszolgálómhoz, hogy letöltsem az új üzeneteimet. Mit ellenőrizzek?*

V A levelezőprogram az SMTP segítségével küldi el az üzeneteket, és egy levéllehívó protokoll (valószínűleg a POP vagy az IMAP) segítségével kéri le a beérkező üzeneteket a kiszolgálóról. A fenti esetben a levéllehívó protokollal lehet átviteli probléma. Sok hálózat más-más kiszolgálókat használ a bejövő és kimenő levelekhez, tehát lehetséges, hogy nem működik a POP- vagy IMAP-kiszolgáló. Keressük meg a levelezőprogramunkban azt a párbeszédablakot, amelyik megadja a POP- vagy IMAP-kiszolgálónk nevét, és adjunk ki visszhangkérést (ping) a kiszolgáló felé, hogy lássuk, válaszol-e.

K *Egy török könyvelőcég 14 számítógépet rendelt a vállalatomtól. Ragaszkodnak hozzá, hogy a számítógépekre telepített levelezőprogramok támogassák a MIME-kódolást. Vajon miért?*

V Az e-mailt eredetileg az angol nyelv karaktereit tartalmazó ASCII-karakterkészlethez tervezték. Az ASCII készletben sok, más nyelvekben használt karakter nem szerepel. A MIME-kódolás nem ASCII-karakterekkel bővíti a karakterkészletet.

Gyakorlat

Ha rendelkezünk internetfiókkal, indítsuk el az elektronikus levelek küldéséhez és olvasásához használt levelezőprogramunkat. Próbáljuk kideríteni, hogy hol található a (kimenő levelekhez használt) SMTP-kiszolgáló, illetve a (bejövő levelekhez használt) POP- vagy IMAP-kiszolgáló beállítása.

Ha igazán kalandvágyók vagyunk, kérdezzük meg egy közeli barátunkat, hogy beállíthatjuk-e a számítógépén a levelezőprogramot úgy, hogy onnan is hozzáférhessünk a postafiókunkhoz. Egyes levelezőprogramok több postafiókot is képesek kezelni, de beállíthatunk egy olyan beépített levelezőprogramot is, amelyet a barátunk nem használ.

Előfordulhat, hogy a barátunk internetszolgáltatójának hálózatáról lekérhetjük a beérkező leveleinket, de kimenő levelet nem küldhetünk onnan a mi internetszolgáltatónk hálózatán található SMTP-kiszolgálónak. Sok internetszolgáltató nem engedi meg, hogy az SMTP-kiszolgálóján keresztül külső e-maileket továbbítsanak.

Ha elvégezzük ezt a gyakorlatot, azt sem szabad elfelejtenünk, hogy bár a levelek lekérése független az internetkapcsolat létrehozásától, a legtöbb levelezőprogram felkínálja a lehetőséget, hogy a levelek lehívásához automatikusan kapcsolódjon az Internetre. A gyakorlathoz mindenképpen élő internetkapcsolattal kell rendelkeznünk.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Fehérlista** – Azoknak a címeknek a listája, ahonnan engedélyezett a levéltovábbítás a tartományba.
- **Feketelista** – Azoknak a kiszolgálóknak a listája, amelyek számára nem engedélyezett, hogy leveleket továbbítsanak a tartományba.
- **Levelezőprogram** – Ügyfélprogram, amely levelek küldésére és fogadására, valamint annak a felhasználói felületnek a kezelésére szolgál, amelyen keresztül a felhasználó műveleteket végez a levelezőrendszeren.
- **Levélfejléc** – Az elektronikus levelek bevezető része, amely információs mezőkből és az azokhoz tartozó értékekből áll.
- **Levéltörzs** – Az elektronikus leveleknek az a része, amelyben az üzenet szövege található.
- **Levélvírus** – Szoftvervírus, amelyet elektronikus levelek mellékleteként terjesztenek.
- **IMAP (Internet Message Access Protocol, internetes üzenet-hozzáférési protokoll)** – Továbbfejlesztett levéllehívó protokoll, amely több, a POP-ban nem elérhető szolgáltatást kínál, például lehetőséget ad rá, hogy a leveleket a kiszolgálóról való letöltés nélkül is elérhessük.
- **MIME (Multipurpose Internet Mail Extensions, többcélú internetes levelezési bővítmények)** – Elektronikus levélformátum, amely kibővíti az internetes levelezés lehetőségeit.
- **POP (Post Office Protocol, postahivatal-protokoll)** – Az Interneten használt egyik népszerű levéllehívó protokoll. A POP lehetővé teszi a felhasználónak, hogy bejelentkezzen egy levélkiszolgálóra, és letöltse vagy törölje a várakozó leveleket.
- **Postafiók** – Az a hely a levélkiszolgálón, ahol egy adott felhasználó beérkező levelei tárolódnak.
- **SMTP (Simple Mail Transfer Protocol, egyszerű levéltovábbítási protokoll)** – A TCP/IP-hálózatokon levélküldésre használt protokoll.
- **Szürkelista** – Azoknak a levélszemétkiszolgáló-észlelő rendszereknek az alapja, amelyek úgy működnek, hogy visszautasítják a kezdeti kézbesítést, hogy lássák, hogy a kiszolgáló újraküldi-e az üzenetet.
- **Webmail** – Rendszer, amely lehetővé teszi a felhasználónak, hogy az elektronikus leveleit egy közönséges webböngészőn keresztül érje el.

19. ÓRA

Adatfolyamok és adatsugárzás

A fejezet tartalmából:

- Folyamprotokollok
- Multimédiás hivatkozások
- Podcasting
- VoIP (Voice over IP)

Az Internetet nem zenelejátszásra vagy régi tévésorozatok nézésére tervezték – az újonnan megjelent adatfolyamokhoz új eljárásokra és protokollokra volt szükség. Ezen az órán az Internet multimédiás eljárásaival ismerkedünk meg.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni az RTP-t és segédprotokolljait.
- El tudjuk magyarázni a szállítási réteg olyan alternatíváit, mint az SCTP és a DCCP.
- Le tudjuk írni, hogyan történik egy multimédiás fájl lejátszása egy webes hivatkozáson keresztül.
- El tudjuk magyarázni, mi az a podcasting, és hogyan működik.
- Be tudunk mutatni néhány fontosabb VoIP-protokollt.

Az adatfolyamok problémája

A felhasználókat elborító hálózati kapcsolatok, átviteli eszközök, képmegjelenítők és PC-hangszórók világában a következő kérdés az volt, hogy az Internet képes lesz-e valaha elavulttá tenni a tévét, a telefont és a rádióállomásokat. Támogathatja például az Internet a hangos kommunikációt? Képesek a szolgáltatók áramló multimédiás programokat közvetíteni a felhasználók kérésére – vagy akár élőben?

A szakemberek és vállalkozók évek óta beszélnek a tévé és a számítógép-rendszerek összeolvasztásáról, de az első modellek nem váltak be, részben a kellő internetes sávszélesség hiánya, részben az otthoni számítógépes hardvereszközök fejletlensége miatt.

A tévét és a számítógépet egyesítő doboz azonban ma már valóság azoknak a felhasználóknak, akik hajlandóak fizetni érte, és az internetes telefonszolgáltatás is egyre terjed. Ezekre a fejleményekre nem kerülhetett volna sor a hardver és az Internet infrastruktúrájának fejlődése nélkül, az igény szerint lehívható multimédiás tartalom új világa azonban a TCP/IP protokollrendszer továbbfejlesztését is megkövetelte.

A multimédiás tartalmak folyamként történő sugárzása (streaming) több kihívást is támaszt a protokollrendszerrel szemben, de a legjelentősebb problémát talán a szolgáltatás minősége okozza. Az Internetet fájlok és véges üzenetek átvitelére tervezték, nem pedig interaktív vagy folyamatos szolgáltatásra. Az adatcsomagok útvonalát az útválasztók határozzák meg, és nincs rá garancia, hogy egységes, folyamatos adatfolyamként érkeznek célba. Az áramló adatsugárzás nagy teljesítményt kíván, méghozzá olyan folyamatossággal, ami biztosítja a hang- és videófolyamok természetességét.

A probléma szemléltetésére vizsgáljuk meg a Szállítási réteg két fő protokollját. Az UDP protokoll gyors, de nem elég sokoldalú vagy megbízható. A TCP protokoll ezzel szemben megbízható, de ezért teljesítménnyel fizetünk. A TCP megbízhatóságát az ellenőrzés és újraküldés rítusa biztosítja, ami viszont bizonytalanságot szül, és ellenkezik a folyamatos adatsugárzás elvével.

Az áramló adatokkal kapcsolatos problémák orvoslására a TCP/IP protokollcsalád több új taggal bővült. Ezen az órán ezek közül ismerkedünk meg az RTP-vel, illetve néhány más folyamprotokollal. Fontos, hogy észrevegyük, hogy az adatfolyamok problémája különböző feladatokat érint, hiszen hangok (FM rádió, VoIP-telefonhívás), videók (élő webközvetítés, igény szerint lehívott filmek), sőt grafikus animációk áramló átvitelére is szükség lehet.

A multimédiás tartalom átvitelére természetesen az is megoldás lehet, ha egyszerűen fájlba mentjük, és a fájlt visszük át e-mailben, webes hivatkozásokon, RSS-sugárzáson vagy egy zenemegosztó alkalmazáson keresztül. Ebben az órában a multimédiás hivat-

kozások működését is megvizsgáljuk, de mivel ezek a megoldások nem igazán különböznek a fájlátvitel más módozataitól, nem ugyanazokkal a kihívásokkal állítják szembe a protokollrendszert, ezért az óra legnagyobb részében az adatfolyamokkal kapcsolatos kérdésekkel foglalkozunk.

RTP (Realtime Transport Protocol)

A megfelelő időben történő, megbízható kézbesítés problémájára számos megoldás született, de az áramló internetes adatfolyamok kirakósjátékának talán az RTP (Realtime Transport Protocol, valósidejű szállítási protokoll) a legfontosabb darabja. Az RTP egy csomagformátumot, valamint egy szabványos módszert határoz meg a hang- és videófolyamok átviteléhez a TCP/IP felett. Neve szerint az RTP szállítási protokoll, de a valóságban ennél kicsit bonyolultabb a helyzet. Az RTP nem váltja fel az elsődleges szállítási protokollokat, hanem az UDP tetejére épül (lásd a 19.1. ábrát), és az UDP-kapuk segítségével éri el az Internetet.

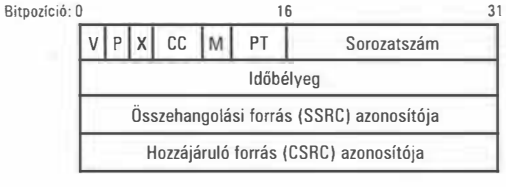
19.1. ábra

Az RTP az UDP segítségével teszi lehetővé az áramló hálózati adatfolyamokat

Bizonyára felmerül bennünk a kérdés, hogy az RTP hogyan oldja meg az UDP-átvitellel kapcsolatos megbízhatósági problémákat. Ahogy a 6. órán megtanultuk, a fejlesztők saját megoldásokkal tehetik megbízhatóvá az UDP-t. Az RTP esetében egy RTCP (Realtime Control Protocol, valósidejű vezérlőprotokoll) nevű társprotokoll figyeli az RTP-munkamenetek szolgáltatásminőségét. Ez lehetővé teszi az alkalmazásnak, hogy igazítson az adatfolyamon – az áramlási sebesség változtatásával vagy esetleg egy kevésbé erőforrásigényes formátumra vagy felbontásra váltással. Ez a megoldás nem küszöböli ki teljesen a problémát, de több lehetőséget ad a csomagok áramlásának figyelésére.

Az RTP-t eredetileg az RFC 1889-ben írták le, de ezt azóta felülírta az RFC 3550. Az RTP-fejléc formátumát a 19.2. ábrán láthatjuk. A fejléc mezői a következők:

- **Változat (Version, V)** – Az RTP változata.
- **Kitöltés (Padding, P)** – Azt jelzi, hogy a csomag tartalmaz-e egy vagy több kitöltő oktettet.
- **Bővítés (Extension, X)** – Azt jelzi, hogy van-e fejlécbővítés.
- **CSRC-szám (CSRC count, CC)** – A rögzített fejlécet követő CSRC-azonosítók száma.
- **Jelölő (Marker, M)** – A kerethatárokat, illetve a csomagfolyam más fontos pontjait jelöli.
- **Értékes tartalom típusa (Payload type, PT)** – Az értékes tartalom formátuma.
- **Sorozatszám** – A munkamenetben elfoglalt helyet jelző szám, amelynek az értéke minden csomag esetében eggyel nő. Ezt a paramétert az elveszett csomagok észlelésére használhatjuk.
- **Időbélyeg** – Az értékes tartalom első oktettjének mintavételi ideje.
- **SSRC** – Egy összehangolási forrást azonosít.
- **CSRC** – A csomag értékes tartalmához hozzájáruló forrásokat azonosítja.



19.2. ábra

Az RTP fejlécformátuma

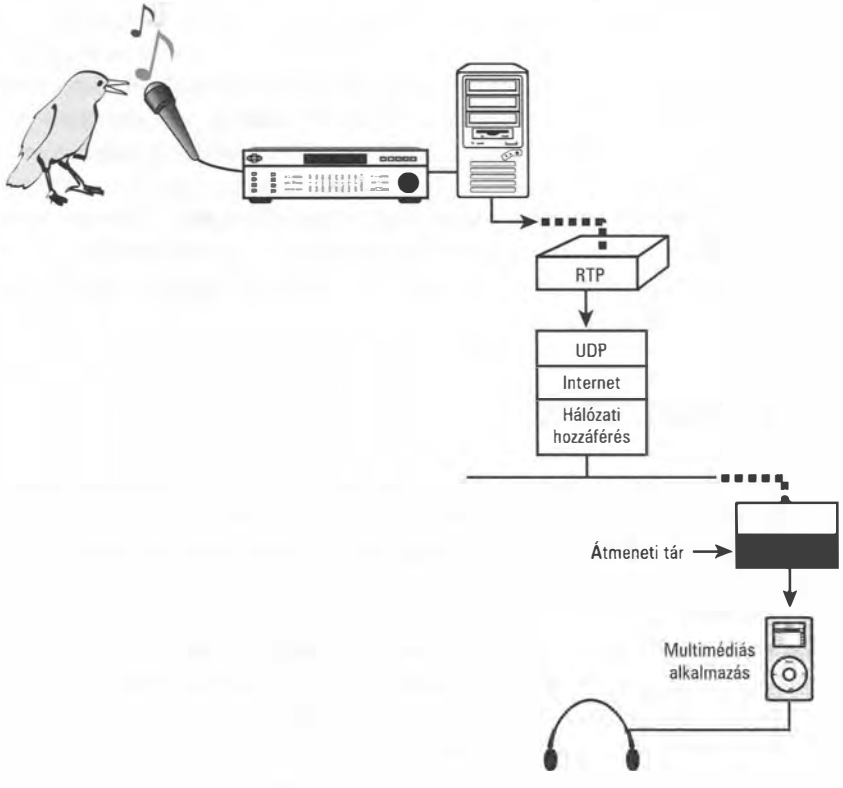
Létezik egy elhagyható RTP-bővítményfejléc is, amely az egyes alkalmazások fejlesztőinek lehetővé teszi, hogy módosításokkal kísérletezve javítsanak a teljesítményen és a szolgáltatás minőségén. Ezenkívül egyes gyártók saját változatot készítettek az RTP-ből; ezeknek az összeegyeztethetősége változó.

Az RTP-t (vagy ha már itt tartunk, bármely más folyamprotokollt) használó audióalkalmazásoknak biztosítaniuk kell valamiféle átmeneti tárolást, hogy a kimenő hangfolyam egyenletes legyen. Az átmeneti tár (buffer) egy memóriablokk, amely a beérkező adatok ideiglenes tárolására szolgál. Az átmeneti tárolás lehetővé teszi az alkalmazásnak, hogy egyenletes ütemben dolgozza fel a bemenetet akkor is, ha az adatok nem egyenletes tempóban érkeznek. Amíg az átmeneti tár nem teljesen üres vagy nem teljesen teli, addig az adatokat fogadó alkalmazás egyenletesen képes feldolgozni a bemenetet.

Az RT protokollcsaládban egy RTSP (Realtime Streaming Protocol, valósidejű adatfolyamprotokoll) nevű protokollt is találunk. Az RTSP olyan parancsokat biztosít, amelyekkel a távoli felhasználók vezérelhetik az adatfolyamot, az RTSP-t tehát úgy kell elképzel-

nünk, mint a tévé távirányítóját. Az RTSP maga nem vesz részt az adatfolyam-sugárzásban, csupán lehetővé teszi a felhasználónak, hogy olyan parancsokat adjon ki a kiszolgálóalkalmazásnak, mint a megállítás, a lejátszás vagy a rögzítés.

Az adatfolyam-sugárzásra a 19.3. ábrán láthatunk egy jellemző forgatókönyvet. Az itt látható esetben hangbemenetet kapunk egy hangfelületen keresztül, amelyet egy számítógépprogramnak átadva digitális formára alakítunk. Az adatfolyam-sugárzó szoftver az adatfolyamot különálló csomagokra bontja, és így viszi át az RTP-n és a TCP/IP-protokollvermen keresztül az adatfolyam-fogadó ügyfélnek, ahol az adatok egy átmeneti tárba kerülnek. A zenelejátszó program innen olvassa ki folyamatosan az adatokat, és küldi a sztereó hangszórókra kimenetként. Mindeközben az RTCP protokoll a munkamenetben részt vevő alkalmazásoknak információkat nyújt a szolgáltatás minőségéről, és ha nem élő előadásról, hanem előre rögzített hang- vagy mozgóképfájlról van szó, az ügyféloldalon levő felhasználó az ügyfélprogramból parancsokat küldhet a kiszolgálónak az RTSP-n keresztül, hogy elindítsa vagy leállítsa az adatfolyamot.



19.3. ábra

Forgatókönyv az adatfolyam-sugárzásra

Átviteli lehetőségek

Annak ellenére, hogy a hang- és videósugárzáshoz széles körben használják az UDP feletti RTP-t, a szakemberek még mindig olyan lehetőségeken törik a fejüket, amelyek a szállítási rétegben kiküszöbölnék a TCP és az UDP alapvető alkalmatlanságát az áramló adatsugárzásra.

Az SCTP (Stream Control Transmission Protocol, adatfolyamvezérlő átviteli protokoll), amelyet az RFC 2000-ben és későbbi dokumentumokban írtak le, egy kapcsolatközpontú szállítási protokoll (ebben a tekintetben tehát a TCP-re hasonlít), de az UDP-től eltérően inkább az üzenetekre összpontosít. Az SCTP arra is lehetőséget ad, hogy egyetlen kapcsolaton keresztül párhuzamosan több üzenetfolyamot tartsunk fenn.

A DCCP (Datagram Congestion Control Protocol, adatcsomagtorlódás-szabályozó protokoll), amelyet az RFC 4340 ír le, szintén kölcsönöz képességeket mind a TCP-től, mind az UDP-től. A DCCP a TCP-hez hasonlóan kapcsolatközpontú, és az UDP-hez hasonlóan gyors, de megbízhatatlan kézbesítést nyújt.

Az SCTP és a DCCP egyaránt végez úgynevezett torlódásszabályozást, de ahogy a nevéből is kiderül, a DCCP-t részben kifejezetten erre a célra tervezték. A torlódásszabályozás a TCP-ben szükséges újraküldések számának csökkentésére és a sávszélesség hatékonyabb kihasználására irányul. A protokoll által alkalmazott algoritmusok az adatfolyam jellemzőinek igazításával teszik optimálissá az áteresztőképességet, és csökkentik az újraküldött csomagok számát. Az SCTP-hez és a DCCP-hez már állnak rendelkezésre megvalósítások. A SCTP valamivel régebbi, és a fejlesztők talán jobban ismerik, de a DCCP kifejezetten ígéretes.

Multimédiás hivatkozások

Nem kell túl sokat szörfölnünk ahhoz, hogy weboldalakba ágyazott videó- és hangfájlokat találjunk. Ha viszont egy hivatkozásra kattintva szöveget vagy zenét hallhatunk, vagy videót nézhetünk, bizonyára kíváncsiak vagyunk rá, hogy ténylegesen mi történik ilyenkor a színfalak mögött.

A válasz természetesen attól függ, hogy hová vezet a hivatkozás. Sok multimédiás hivatkozás egyszerűen egy fájlra mutat. ahogy korábban, a 17. fejezetben megtanultuk, egy másik erőforrásra az `<a>` címke `HREF` jellemzőjével hivatkozhatunk. A korábbi példákban az erőforrás egy weboldal volt, a hivatkozás azonban bármilyen típusú fájlra mutathat, amíg a böngésző tudja, hogyan kell értelmeznie a fájl tartalmát. A mai böngészők nagyon sokféle fájlformátumot képesek kezelni. Windows rendszereken a fájlkiterjesztés (a fájlnévnek a pont utáni része – például `.doc`, `.gif` vagy `.avi`) árulja el a böngészőnek (vagy az operációs rendszernek), hogy melyik alkalmazással kell megnyitnia a fájlt.

Más operációs rendszerek a fájltípust a kiterjesztéstől függetlenül határozzák meg. Amennyiben a böngésző számítógép rendelkezik a videó- vagy hangfájl megnyitásához szükséges szoftverrel, és ha a böngészőt vagy az operációs rendszert úgy állították be, hogy felismeri a fájlt, a weboldal egy közönséges hivatkozáson keresztül hivatkozhat a fájlra, és a böngésző számítógép megnyitja vagy végrehajtja a fájlt, amikor a hivatkozásra kattintanak.

Az elterjedtebb videófájl-formátumok a következők:

- .AVI (Audio Visual Interleave) – A Microsoft által kifejlesztett hang- és videóformátum.
- .MPEG (Motion Picture Experts Group) – Népszerű, magas minőségű digitális videóformátum.
- .SWF – Képernyőn megjelenített animációkhoz és Flash-videókhoz használt formátum.
- .MOV (QuickTime) – Az Apple formátuma, amelyet eredetileg a Macintosh rendszerekhez fejlesztettek ki, de a QuickTime sok más rendszerhez is elérhető.

A YouTube többféle formátumot elfogad, de a legtöbb videót FLV típusú, `.swf` fájlba beágyazott Flash-videóvá alakítja át, mert a Flash-formátum gyors lejátszást tesz lehetővé, és a Flash-lejátszó széles körben elérhető. A hangfájlformátumokból is több áll rendelkezésre az Interneten, de a jogvédett MP3 formátum messze a legnépszerűbb a zenefájlok letöltéséhez és lejátszásához készítettek közül.

Amikor multimédiás alkalmazást telepítünk az ügyfélgépre (például amikor a QuickTime-megjelenítőt telepítjük), a telepítőprogram általában bejegyzi azokat a fájlkiterjesztés(eke)t, amelyeket látva a számítógépnek az adott alkalmazást kell elindítania. Egyes esetekben, ha a megfelelő program vagy bővítmény nem érhető el a fájl lejátszásához, a rendszer a felhasználót egy letöltési oldalra irányítja, és a program önműködően települ.

A multimédiás fájlok rögzítésével, kódolásával és megtekintésével kapcsolatban természetesen még sokkal több mindenről lehetne szót ejteni, de a részletek nem a HTTP-re vagy a TCP/IP-re tartoznak. A hálózat szemszögéből a böngésző egyszerűen letölt egy fájlt, amikor a felhasználó a hivatkozásra kattint.

Az a tény, hogy a böngésző néha más alkalmazások segítségével nyitja meg és hajtja végre a fájlokat, jól jelzi, hogy az egész HTTP-rendszer (a HTTP, a HTML, a webkiszolgáló és a webböngésző) a lényegét tekintve kézbesítési módszer, hasonlóan az alatta található TCP/IP-rétegekhez.

A hivatkozás néha egy olyan valódi multimédiás adatfolyamhoz való csatlakozást tesz lehetővé, amilyenekről ebben a fejezetben beszéltünk. Az Interneten található adatfolyam-sugárzó kiszolgálók kérésre (on demand, igény szerint) sugározzák az áramló hang- és videótartalmat a hivatkozásra kattintó felhasználó gépére.

Az adatfolyam-sugárzás kezdeményezése egy webböngészőn keresztül gyakran az óra korábbi részében megismert RSTP protokoll segítségével történik. Ahogy már említettük, az RSTP maga ténylegesen nem vesz részt az adatfolyam-sugárzásban, csupán vezérlőrendszert biztosít az adatfolyam elindításához és leállításához. Egy olyan URL, mint az `rstp://greatmovies.com/casablanca.mp4`, például egy Bogart-klasszikust sugározhat az Asztalunkra – amennyiben a böngésző rendelkezik a megfelelő szoftverrel a kapcsolat feldolgozásához.

A helyzetet némileg bonyolítja, hogy az adatfolyamokat néha webes parancsfájlok fedik el, vagy szándékosan elrejtik őket a szemünk elől. Előfordul, hogy egy multimédiás adatfolyam URL-jét valójában egy apró szövegfájl, egy úgynevezett metafájl rejti. A címsávban hivatkozott erőforrás erre a metafájlra mutathat, amelynek a kiterjesztése `.pls`, `.ram`, `.asx`, `.wax`, `.wvx` stb. lehet. Ha kíváncsiak vagyunk, hová vezet a hivatkozás, az Interneten számos segédprogramra lelhetünk, amelyek segítenek megtalálni a rejtett multimédiás adatfolyamok helyét.

Podcasting

A letölthető multimédiás fájlok és az igény szerint sugárzott áramló adatfolyamok között helyezkedik el egy (legalábbis elméletben különálló) lény, amit podcast-nak hívnak. A podcasting az Apple híres iPod eszközéről kapta a nevét, de ma már általánosabb értelemben használják.

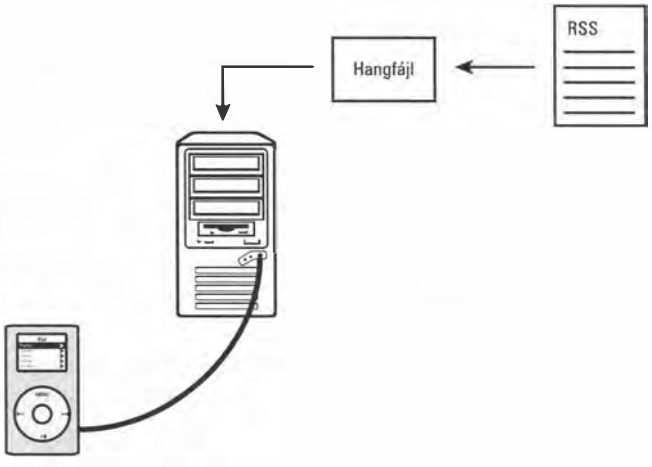
Ha előfizetünk egy podcast-ra, egy RSS-folyamon keresztül kapunk multimédiás (általában audió) tartalmat. Az RSS-t eredetileg hírszolgáltató csatornának tervezték – valahogy úgy működik, mintha az Interneten keresztül kapnánk meg a reggeli újságot. A felhasználó előfizet egy RSS-hírszolgáltatásra, és a legfrissebb hírek automatikusan megérkeznek az Asztalára. A lényeg, hogy a felhasználónak nem magának kell híreket keresgélnie egy webhelyen, hanem ha az előfizetés hatályba lépett, az új cikkek önműködően letöltődnek az olvasó gépére (lásd a 19.4. ábrát).

A podcasting célja multimédiás fájlok sugárzása a felhasználó számára közvetlenül az RSS eszközeinek a segítségével. Az RSS ugyanis képes fájlokat mellékelni a hírekhez, és ezek a mellékletek szállítják a podcast adatfolyamát.

A podcast-ügyfélprogramok a podcast-fájlok kezelését végzik, és értesítenek a frissítésekről. Az iTunes felhasználói könnyen hozzáférhetnek podcast-folyamokhoz, és más zenelejátszók is kínálnak hasonló szolgáltatást. Az iPodder egy nyílt forrású podcast-ügyfél, amely Windows, MacOS, Linux és BSD rendszeren is képes működni.

A podcasting-nak persze csak akkor van értelme, ha a tartalmat rendszeresen frissítik, ami azt jelenti, hogy bárki is állítja elő az adatfolyamot a kiszolgálóoldalon, valamiféle folyamatos programot kell sugároznia. A podcast-ok szerte a világon nagy népszerű-

ségre tettek szert: az RSS csodája lehetővé teszi, hogy az előfizetők interjúkat vagy oktató előadásokat hallgassanak, illetve zenés vagy vicces videókat nézzenek, ugyanúgy, mintha rádiót hallgatnának, vagy tévét néznének.



19.4. ábra

A podcasting multimédiás fájlokat továbbít egy RSS-szolgáltatáson keresztül

Hangátvitel IP felett (VoIP)

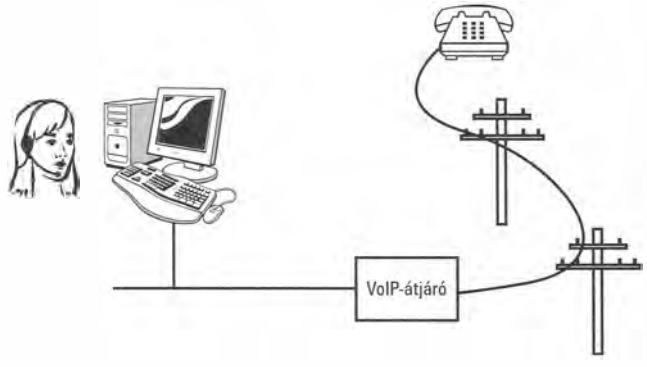
Az internetes telefonálás egyes területeken ma már hétköznapinak számít, ráadásul a TCP/IP-telefonszolgáltatás gyakran olcsóbb és sokoldalúbb, mint a hagyományos. Az internetes telefonhívások sok szempontból csupán az áramló hangfolyamok egy másik formájának tekinthetők, ezért nem meglepő, hogy a VoIP- (Voice over IP, hang- vagy beszédátvitel IP felett) kommunikációban a beszéd átvitelére az RTP a legnépszerűbb protokoll. Maga a beszéd azonban csak a kirakósjáték egyik darabja. A keresett felhasználó megtalálása, a híváskezdeményezés, valamint a munkamenet elindítása és elegáns befejezése új eszközöket és protokollokat igényel. Ezen kívül, ha az IP-telefonszolgáltatásunkat össze akarjuk kapcsolni a hagyományos telefonhálózattal, az is gondot jelent, hogy olyan vezérlőrendszert biztosítsunk, amely összeegyeztethető (vagy legalábbis összekapcsolható) a hagyományos telefonrendszerekben használt vezérlőkkel.

Az IP-telefonálás egy (a normál telefonokhoz hasonló, de a TCP/IP-vel való működésre tervezett) fizikai telefonkészüléken vagy egy úgynevezett szoftveres telefonon keresztül történhet. Az utóbbi egy olyan, a telefon szerepét betöltő számítógépprogram, amely hangbemenetet fogad egy mikrofonról, hangkimenetet küld a hangszórókra vagy a fejhallgatóra, és a világgal a számítógép TCP/IP hálózati szoftverén keresztül tartja a kapcsolatot. A telefon mindkét esetben olyan jeleket küld a hálózaton át, amelyeket a hívás másik végén egy másik telefonnak kell fogadnia és értelmeznie.

A VoIP-telefonhívások kezdeményezésére és kezelésére számos protokoll létezik. Az International Telecommunication Union (Nemzetközi Telekommunikációs Unió) H.323 protokollrendszere egy nagy protokollcsalád, amely a VoIP kezelésére, konferenciahívások bonyolítására és más kommunikációs feladatokra is alkalmas. Sok VoIP-rendszert a H.323-hoz terveznek.

Egy másik, újabb és egyszerűbb (és könnyebben leírható) protokoll az SIP (Session Initiation Protocol, munkamenet-kezdeményező protokoll). Az SIP protokoll az alkalmazásrétegben működik, és egy kommunikációs munkamenet elindítására, leállítására és kezelésére szolgál. Az SIP egy úgynevezett meghívót küld egy távoli felhasználónak, amely a VoIP környezetében a híváskezdeményezésnek felel meg. A hívások kezdeményezésén és befejezésén túl az SIP olyan szolgáltatásokat is nyújt, mint a konferenciabeszélgetés, a hívástovábbítás, illetve a *szolgáltatás-egyeztetés*. A kapcsolat létrejöttekor a tényleges áramló beszédátvitel egy olyan protokoll segítségével történik, mint az RTP.

Az IP-telefonálással kapcsolatban a másik problémát a felhasználóknak a régimódi vezetékeken keresztüli elérése jelenti. Az Internet és a telefonhálózat között egy VoIP-átjáróeszköz szolgál felületként (lásd a 19.5. ábrát). A VoIP-felhasználók közvetlenül az Interneten át társaloghatnak egymással, és nincs szükség átjáróra, de ha egy olyan számot hívnak, amely a hagyományos telefonhálózatba tartozik, a hívás egy VoIP-átjáróeszközhöz kerül. Az internettelefon-felhasználóknak egy VoIP-átjárószolgáltatásra kell előfizetniük, hogy hozzáférést szerezzenek egy ilyen átjáróhoz. Ez általában a VoIP-telefonszerződés része, de az átjárón keresztüli kapcsolás költsége általában jóval magasabb a csak internetes végpontok közötti hívásokénál. Az internetes végpontok közötti telefonálás gyakran a világ bármely pontjára ingyenes (vagy szinte ingyen van) a havidíjat fizető előfizetőknek.



19.5. ábra

A VoIP-átjárók felületként szolgálnak a hagyományos telefonhálózathoz

Összefoglalás

Ezen az órán néhány olyan technológiával ismerkedtünk meg, amely áramló multimédiás adatfolyamok sugárzását teszi lehetővé az Interneten. Tanultunk az RTP-ről, az RSTP-ről és az RTCP-ről, valamint megismertük az SCTP és DCCP szállítási protokollokat, és megvizsgáltuk, hogyan játszhatunk le zenét és videót egyetlen egérkattintással a multimédiás hivatkozásokon keresztül. Ezenkívül szót ejtettünk a podcasting-ról, és az órát az IP feletti hangátvitelre (VoIP) vetett pillantással zártuk.

Kérdezz-felelek

K *Miért alkalmatlanok a Szállítási réteg elsődleges protokolljai az áramló adatátvitelre?*

V Az UDP gyors, de megbízhatatlan, a TCP pedig – bár megbízható – olyan szabályozókat használ a kézbesítés vezérlésére, amelyek lassúvá teszik, és újraküldést tehetnek szükségessé.

K *Mi a feladata az RTP két testvérprotokolljának, az RTCP-nek és az RTSP-nek?*

V Míg az RTP az áramló átvitelről gondoskodik, az RTCP a szolgáltatás minőségét figyeli és jelzi, az RTSP-t pedig olyan vezérlőparancsok kiadására használjuk, amelyekkel elindíthatjuk és leállíthatjuk az adatfolyamot.

K *Miért alakítja át a YouTube a benyújtott videókat Flash-formátumra?*

V A Flash hatékony és megbízható videóformátum, és a Flash-lejátszó széles körben elérhető.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- DCCP (Datagram Congestion Control Protocol, adatcsomagtorlódás-szabályozó protokoll) – Alternatív protokoll a Szállítási rétegben az adatfolyam alapú alkalmazásokhoz.
- Podcasting – RSS-folyamon keresztül sugárzott multimédiás fájlok kézbesítésére szolgáló módszer.
- RTCP (Realtime Control Protocol, valósidejű vezérlőprotokoll) – Az RTP számára a szolgáltatás minőségét figyelő protokoll.
- RTP (Realtime Transport Protocol, valósidejű szállítási protokoll) – Népszerű folyamprotokoll.
- RTSP (Realtime Streaming Protocol, valósidejű adatfolyam-protokoll) – Az RTP számára vezérlőparancsokat biztosító protokoll.
- SCTP (Stream Control Transmission Protocol, adatfolyamvezérlő átviteli protokoll) – Alternatív protokoll a Szállítási rétegben az adatfolyam alapú alkalmazásokhoz.

- **SIP (Session Initiation Protocol, munkamenet-kezdeményező protokoll)** – A VoIP-kommunikáció kezelésére szolgáló protokoll.
- **Szolgáltatás-egyeztetés** – Alkalmazások vagy eszközök között zajló egyeztetés, amelynek a célja a kapcsolathoz használt szolgáltatások közös halmazának meghatározása.
- **VoIP (VoIP, hangátvitel IP felett)** – TCP/IP-hálózaton keresztül nyújtott telefonszolgáltatás.

VI. RÉSZ

Haladó témák

20. ÓRA

Webszolgáltatások

A fejezet tartalmából:

- Webszolgáltatások
- XML
- SOAP
- WSDL
- Webes tranzakciók

A webes technológiák új forradalmat idéztek elő a szoftverfejlesztésben. A webszolgáltatások felépítése lehetővé teszi a programozónak, hogy a Web eszközeit olyan bonyolult feladatok végrehajtására használja fel, amilyenekre a HTML megalkotói soha nem gondoltak volna. Ezen az órán a webszolgáltatások rendszerét vesszük górcső alá, valamint röviden bemutatjuk, hogy az e-kereskedelmi webhelyek hogyan dolgozzák fel a webes tranzakciókat.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni a webszolgáltatások felépítését.
- Érteni fogjuk az XML, az SOAP és a WSDL szerepét a webszolgáltatások működésében.
- El tudjuk magyarázni, hogy az e-kereskedelmi webhelyek hogyan bonyolítják le a pénzügyi tranzakciókat.

A webszolgáltatások működése

Most, hogy szinte minden számítógép rendelkezik webböngészővel, és a webkiszolgálók működését is széles körben értik, a szoftverfejlesztők és a jövőt fürkészők új módokat igyekeznek kieszelni a Web eszközeinek használatára. Régen, ha egy programozó hálózati alkalmazást akart írni, akkor készítenie kellett egy egyéni kiszolgálóprogramot, valamint egy egyéni ügyfélprogramot, és ki kellett dolgoznia egy egyéni nyelvtant vagy formátumot is, amelynek a segítségével a két alkalmazás információt cserélhetett. A szükséges szoftver megírása rengeteg időt és agykapacitást emésztett fel, de a számítógép-hálózatok jelentőségének növekedésével az adatbeágyazás és a központosított kezelés igénye ügyfél-kiszolgáló alkalmazásokat kívánt. Természetesen léteztek hálózati programfelületek – másképp sok, ebben a könyvben leírt klasszikus alkalmazás sem születhetett volna meg –, de a hálózati programozás jellemzően a hálózati felület hosszadalmas, magas költségű kódolását követelte meg.

Idővel egyszerűbb megoldás született: a Web meglevő eszközeinek, technológiáinak és protokolljainak használata az egyéni hálózati alkalmazások alapjaként. Ezt a megközelítést, amelyet a nyílt forrás hívei és a fejlesztőeszközök gyártói mellett olyan nagyvállalatok támogattak, mint az IBM és a Microsoft, ismerjük *webszolgáltatási architektúraként*.

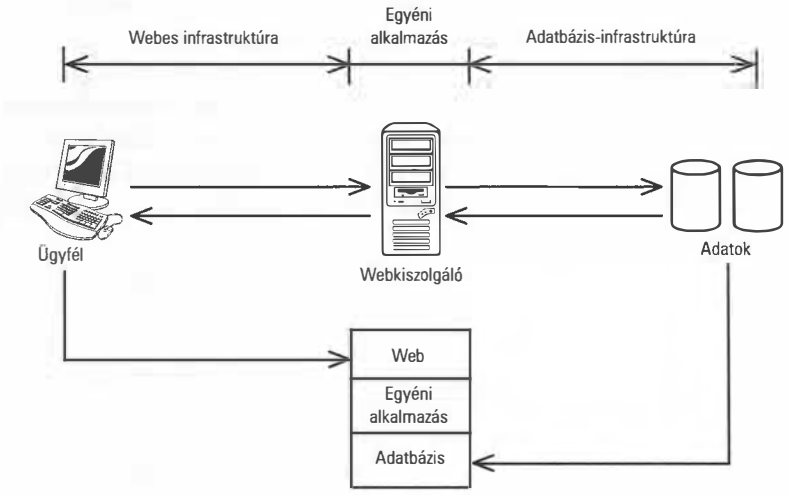
A webszolgáltatási architektúra elve az, hogy a webböngésző, a webkiszolgáló és a TCP/IP-protokollverem kezeli a hálózati kapcsolat részleteit, hogy a programozó az alkalmazás működésére összpontosíthasson. Az utóbbi években ez a technológia meghaladta a Web mint a globális Internet arca eredeti elképzelését – a webszolgáltatási architektúrát ma már bármilyen hálózati alkalmazás építésére alkalmas megoldásnak tekintik, függetlenül attól, hogy az alkalmazás ténylegesen kapcsolódik-e az Internetre. Az olyan nagy és befolyásos gyártók, mint a Sun, a Microsoft vagy az IBM, rengeteg erőforrást áldoznak arra, hogy összetevőket fejlesszenek a webszolgáltatások e rendszerének támogatására.

Amit a webszolgáltatások rendszereként ismerünk, annak a HTTP kézbesítési rendszer csupán egy része. Ugyanilyen jelentőséggel bírnak azok az összetevő-architektúrák is, amelyek kész osztályokat, függvényeket és programozási felületeket nyújtanak a web alapú környezetben végzett munkához.

Webszolgáltatásokat gyakran használnak olyan esetekben, amikor egyszerű ügyfélkapcsolatra van szükség egy kiszolgálóhoz, amely valamilyen nyilvántartást vezet, vagy rendeléseket dolgoz fel. Egy gyártócég például egy webszolgáltatásként működő program segítségével rögzítheti a megrendeléseket, követheti nyomon a szállítást, és tarthatja naprakészen a raktárnyilvántartást.

Szinte minden nagyobb cégnek szüksége van valamilyen szoftverre, amellyel nyomon követheti az üzleti tárgyalásokat, a rendeléseket és a raktárkészletet. A különböző szolgáltatások és tranzakciók egyetlen, egységes környezetben való egyesítésére egy webszolgáltatási keretrendszer kiválóan alkalmas.

A 20.1. ábrán egy teljes webszolgáltatási környezetet láthatunk. Az előtérben (az ábra bal oldala) a programozó a meglevő webes infrastruktúrára támaszkodhat, amely az adatátvitelt kezeli, valamint felhasználói felületet nyújt az ügyfélszámítógép webböngésző alkalmazásán keresztül, a háttérben pedig egy szintén már meglevő adattárolási rendszerre, amelyet egy SQL-adatbázis biztosít. A programozónak így csak az ábrán látható középső részre kell összpontosítania – ahol a webszolgáltatások rendszerének készen kapott összetevői tovább egyszerűsítik a programozást.



20.1. ábra

A webszolgáltatások programozási modellje

Az adatok XML formátumban haladnak át a webszolgáltatások rendszerének összetevőin. Az XML hatékony és általános formátum, amelyek értékek jellemzőkhöz rendelését teszi lehetővé. A szakemberek hamar rájöttek, hogy a rendszer még jobban működik, ha a szolgáltatások meghívására, illetve a válaszok előállítására és átvitelére a hálózaton az XML formátumot használják. Az XML formátumú adatok átadására a webszolgáltatási folyamatok között az SOAP (Simple Object Access Protocol, egyszerű objektumelérési protokoll) nyújt szabványos módot. Az SOAP ezen kívül azt is leírja, hogy miként használható az XML és a HTTP távoli eljárások meghívására. Az óra későbbi részében majd látni fogjuk, hogy az SOAP-üzenetek a webszolgáltatás-leíró nyelven (WSDL, Web Services Description Language) meghatározott hálózati szolgáltatások között haladnak oda-vissza.

XML

Amint a felhasználók, a gyártók és a webtervezők hozzászoktak a HTML-hez, máris többre vágytak. A kiszolgáló- és ügyféloldali programozási eljárások fejlődése arra késztetett sok szakembert, hogy elgondolkodjon, vajon nem lehetne-e valahogy kibővíteni a HTML merev elemrendszerét. A céljuk az volt, hogy túllépjenek a jelölőnyelvnek mint szöveg és grafika formázására szolgáló módszernek a fogalmán, és a nyelvet egyszerűen *adatok* átvitelére használják. A töprengés eredménye egy új jelölőnyelv lett, amelyet eXtensible Markup Language-nek (bővíthető jelölőnyelv), röviden XML-nek neveztek el.

Ahogy az óra korábbi részében megtanultuk, a HTML formátumú adatok jelentése és környezete arra korlátozódik, amit az előre meghatározott HTML-elemek (címkék) segítségével ki tudunk fejezni: ha az adat `<H1>` címkék között áll, akkor címsorként értelmezendő, ha pedig `<A>` címkék zárják közre, akkor hivatkozásról van szó. Az XML ezzel szemben lehetővé teszi, hogy a felhasználó saját elemeket határozzon meg. Az adatok jelentése az lehet, amit csak jelezni szeretnénk velük, és mi találhatjuk ki, hogy az adatok jelölésére milyen címkéket használunk. Például ha szeretjük a lóversenyt, akkor létrehozhatunk egy XML fájlt, amely a kedvenc lovainkról tárol információkat. Ebben a fájlban az alábbiakhoz hasonló bejegyzések lehetnek:

```
<lovak>
      <lo_neve="Winky" fajta="telivér">
            <nem="mén" />
            <kor="3" />
      </lo>
       <lo_neve="Istennő" fajta="arabs">
            <nem="kanca" />
            <kor="3" />
      </lo>
      <lo_neve=""Gecko" fajta="ismeretlen">
            <nem="mén" />
            <kor="14" />
      </lo>
</lovak>
```

Az XML formátum némileg hasonlít a HTML-re, de nyilvánvaló, hogy nem HTML. (El tudjuk képzelni, milyen zavarba jönne a böngészőnk, ha olyasmit próbálnánk neki HTML-címkeként átadni, mint a `<lo_neve>?`) Az XML-ben tetszőleges címkéket használhatunk, mert az adatokat nem egy olyan konkrét, mereven behatárolt alkalmazás számára készítjük elő, mint a webböngésző – az adatok csupán adatok. Az elv az, hogy aki létrehozta a fájl szerkezetét, később készít egy alkalmazást vagy stíluslapot, amely képes elolvasni a fájlt, és értelmezni a benne levő adatokat.

Az XML rendkívül hatékony eszköz adatok alkalmazások közötti átadására. Egy parancsfájl vagy egy saját készítésű alkalmazás egyszerűen előállíthat XML fomátumú kimenetet, illetve könnyen elolvashatja az XML-ként érkező bemenetet. Annak ellenére, hogy a bön-

gészők nem képesek közvetlenül elolvasni, az XML-t széles körben használják a Weben. Egyes esetekben az XML-adatokat a kiszolgálóoldalon állítják elő, majd megjelenítésre alkalmas HTML-lé alakítják, mielőtt átadnák a böngészőnek. Egy másik megoldás, amikor egy kísérő állományt, egy úgynevezett rangsorolt vagy lépcsőzetes stíluslapot (CSS, Cascading Style Sheet) mellékelnek, amely elárulja, hogyan kell értelmezni és megjeleníteni az XML-adatokat. Az XML azonban nem korlátozódik a Webre. A programozók ma már más környezetekben is alkalmazzák, ahol csak egyszerű, kényelmes formátumra van szükség értékek jellemzőkhöz rendeléséhez.

Az XML tehát adattárolási és -átviteli eszközként messze túlmutat a Világhálón. Amíg az XML-adatokat író és az adatokat olvasó alkalmazás meg tud egyezni az elemek jelentésében, az adatokat könnyen és gazdaságosan lehet átvinni a programok között – és mindezt a varázslatos XML teszi lehetővé.

Az XML-re gyakran mondják, hogy „jelölőnyelv jelölőnyelvek létrehozásához".

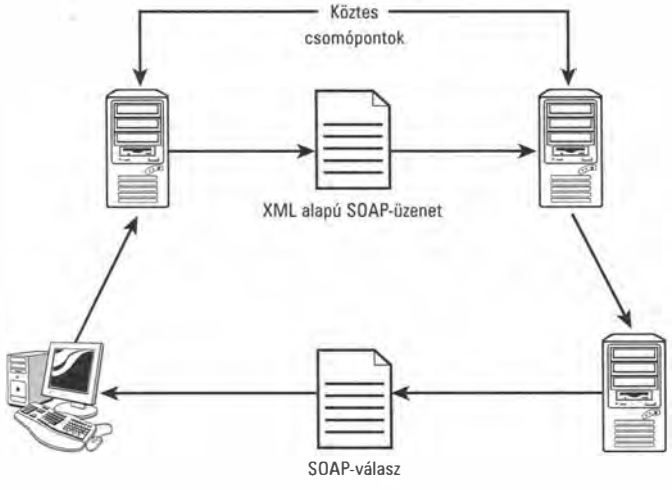
SOAP

Az XML egy általános formátumot határoz meg az alkalmazásadatok cseréjéhez. Az univerzális XML-leírás azonban önmagában nem elég ahhoz, hogy a fejlesztők egyszerű és elegáns webszolgáltatásokat készíthessenek. Bár az XML hatékony formátum a programadatok olvasásához és írásához, önmagában nem nyújt szabványos formátumot az adatok szervezéséhez és értelmezéséhez. Ezt a szerepet az SOAP szabvány látja el. Az SOAP a webszolgáltatás ügyfele és kiszolgálója között átadott XML alapú üzenetek cseréjének szabványos protokollja.

Az SOAP-t úgy tervezték, hogy a kommunikációt úgynevezett SOAP-csomópontok között bonyolítsa. (Az SOAP-csomópontok lényegében olyan számítógépek vagy alkalmazások, amelyek támogatják az SOAP-t.) Az SOAP leírása azoknak az üzeneteknek a szerkezetét határozza meg, amelyeket az SOAP-küldő az SOAP-fogadónak küld. Útközben az üzenet köztes csomópontokon is áthaladhat, amelyek valamilyen szempontból feldolgozzák az információt (lásd a 20.2. ábrát). Egy köztes csomópont végezhet például naplózást, vagy valamilyen módon módosíthatja az üzenetet, mielőtt az elérné a végcélját.

Elvben egy az ügyféltől származó SOAP-üzenet azt mondja, hogy „Íme néhány bemenő adat; dolgozd fel ezt a bemenetet, és küldd el nekem a kimenetet.". Az alkalmazás működése ilyen XML alapú SOAP-üzenetek sorozatából áll, amelyekben a végpontok információkat küldenek, és válaszokat kapnak rájuk. Az SOAP-üzenetek formális szerkezete lehetővé teszi a szoftverfejlesztőnek, hogy egyszerűen elkészítsen egy SOAP

alapú ügyfélprogramot, amely együttműködik a kiszolgálóval. Egy autókölcsönző cég például, amelynél egy webes kiszolgálóprogramon keresztül lehet kocsit rendelni, egyszerűen a fejlesztő rendelkezésére bocsáthatja azokat a leírásokat, amelyek egy olyan egyéni ügyfélprogram elkészítéséhez szükségesek, amely képes kapcsolódni a kiszolgálóhoz, és lefoglalni egy autót.



20.2. ábra

Az SOAP-üzenetek a küldő és a fogadó között köztes csomópontokon is áthaladhatnak

Az SOAP-üzenetek egy elhagyható fejlécből és egy üzenettörzsből állnak. A fejléc olyan címkéket, meghatározásokat és metaadatokat tartalmaz, amelyekre az üzenet útvonalán található csomópontoknak van szükségük, a törzs pedig az üzenet címzettjének szóló adatokat tárolja. Az autókölcsönző szolgáltatás esetében például az üzenettörzs olyan adatokat tartalmazhat az ügyféltől, amelyek a kibérelni kívánt autót írják le, illetve azt, hogy a kocsinak mikor kell rendelkezésre állnia.

WSDL

A webszolgáltatás-leíró nyelv (Web Services Description Language, WSDL) a webszolgáltatásként működő alkalmazásokhoz társuló szolgáltatások leírására biztosít XML-formátumot. A W3C WSDL-szabványa szerint „a WSDL egy XML-formátum, amellyel hálózati szolgáltatások írhatók le dokumentum- vagy eljárásközpontú információkat tartalmazó üzeneteken végzett műveleteket végrehajtó végpontok halmazaként". A WSDL tehát az SOAP-üzeneteken keresztül információcserét folytató szolgáltatások meghatározására szolgáló formátum.

A WSDL-dokumentumok elsősorban meghatározásokból állnak. A dokumentumban található meghatározások az átvitt adatokkal, az azokon végrehajtandó műveletekkel, valamint a szolgáltatáshoz kapcsolódó egyéb adatokkal és a szolgáltatás helyével kapcsolatos információkat írnak le.

A WSDL nem korlátozódik az SOAP-re – más webszolgáltatási kommunikációs protokollokkal együtt is használható. Egyes esetekben a WSDL-t közvetlenül a HTTP-vel használják, hogy egyszerűsítsék a szolgáltatás felépítését, és a műveleteket a HTTP lelkét jelentő alapvető `GET`- és `POST`-stílusú műveletekre korlátozzák.

Webszolgáltatási vermek

Az XML-lel, az SOAP-vel, a WSDL-lel és a TCP/IP, illetve a webszolgáltatási keretrendszerek háttérben levő összetevőivel felfegyverkezve a fejlesztők gond nélkül készíthetnek könnyű és egyszerű ügyfél- és kiszolgálóprogramokat, amelyek egy webes felületen keresztül kommunikálnak. Magához a TCP/IP-hez hasonlóan a webszolgáltatási környezetek összetevői is egy vermet alkotnak. A főbb szoftvergyártók saját webszolgáltatási vermeket nyújtanak az ügyfeleiknek. A teljes rendszer egy csomag, amely kiszolgálószoftverből, fejlesztőeszközökből, sőt akár hardvereszközökből áll, és tanácsadási szolgáltatások, illetve néha megrendelésre készített, egyéni alkalmazások társulnak hozzá.

A Linux-gyártók és -fejlesztők gyakran beszélnek a LAMP veremről, amely olyan nyílt forrású összetevők gyűjteménye, amelyeket könnyen hozzá lehet igazítani a webszolgáltatási környezetekhez. A megjegyezhető LAMP betűszó a verem főbb elemeit jelöli:

- **Linux** – A kiszolgálórendszeren futó kiszolgálóalkalmazásokat támogató operációs rendszer.
- **Apache** – Az XML alapú SOAP-üzeneteket szolgáltató webkiszolgáló.
- **MySQL** – A háttérben levő adatszolgáltatásokhoz hozzáférést nyújtó adatbázisrendszer.
- **PHP (vagy Perl, vagy Python)** – Egy „webkész" programozási nyelv, amelyen a webszolgáltatásként működő egyéni alkalmazás kódját írják.

A jogvédett webszolgáltatási infrastruktúrák hasonló szolgáltatásokat nyújtanak. A Java nyelvet gyakran használják webszolgáltatásokhoz – nem csak a Sun (a Javát megalkotó cég), hanem az IBM WebSphere és más rendszerek is. A Microsoft a .NET keretrendszer eszközein keresztül biztosít a Javához hasonló megoldásokat.

E-kereskedelem

Az e-kereskedelmi webhelyek nem szükségszerűen az óra eddigi részében felvázolt webszolgáltatási architektúra megvalósításai, de ettől függetlenül használhatnak bizonyos webszolgáltatási eljárásokat, különösen a háttérben. Az e-kereskedelem jó példája annak, hogy miként lehet az alkalmazásokat és összetevőket egyesíteni a Web eszközeivel.

A gyártó- és hirdetőcégek hamar felismerték, hogy a Web kiválóan alkalmas arra, hogy rávegyék az embereket a vásárlásra. Nem titok, hogy sok webhely látszatra szinte másból sem áll, mint egymásba fonódó reklámok tömegéből. A túlzásba vitt – és inkább kétséget ébresztő – reklámok ellenére azonban tény, hogy a Weben valóban kényelmesen és költséghatékonyan lehet vásárolni. A kereskedő pedig ahelyett, hogy katalógusok ezreit küldené ki postán, egyszerűen felteheti a katalógust a Webre, és a vásárlókra hagyhatja, hogy keresés és hivatkozás útján rábukkanjanak.

A webes vásárlás addig nem igazán vált jövedelmező üzletté, amíg a szoftvergyártók meg nem oldották azokat a biztonsági problémákat, amelyeket a hitelkártya-adatoknak a nyílt Interneten keresztüli elküldése okoz. Valójában az internetes eladás nem is lenne lehetséges biztonságos hálózati eljárások nélkül. Ma már azonban a legtöbb böngésző képes biztonságos kommunikációs csatornát nyitni a kiszolgálóval. Ez a biztonságos csatorna lehetetlenné teszi a kibertolvajok számára, hogy elfogják a jelszavakat vagy a hitelkártya-adatokat.

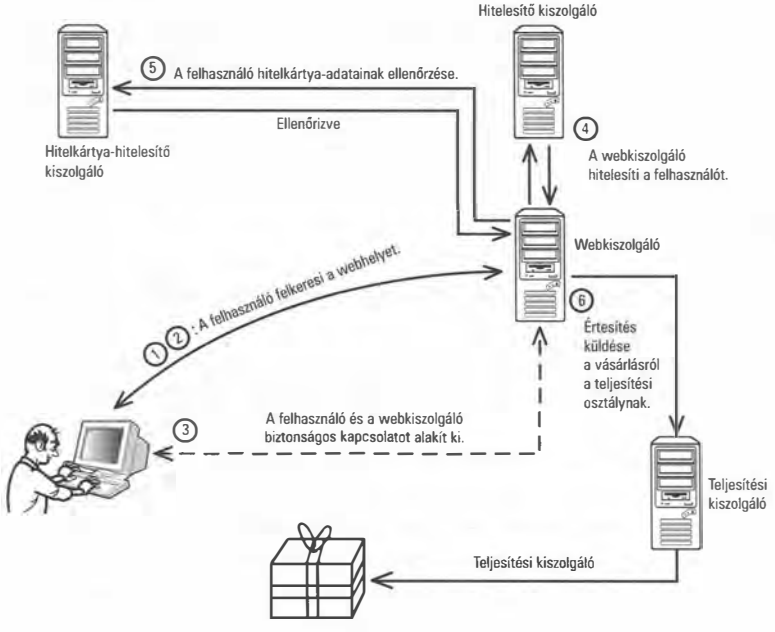
A webes tranzakciók jellemző forgatókönyvét a 20.3. ábrán láthatjuk. Az eljárás menete a következő:

1. Egy webkiszolgáló online katalógust nyújt, amely a Weben érhető el. A felhasználó az Interneten keresztül egy távoli helyről böngészhet a kínált termékek között.
2. A felhasználó úgy dönt, hogy megvásárol egy terméket, és a Buy This Product (Termék megvásárlása) hivatkozásra kattint a weboldalon.
3. A böngésző és a kiszolgáló biztonságos kapcsolatot létesít. (Az SSL-ről és más biztonságos kommunikációs eljárásokról a 23. órában beszélünk bővebben.) Ekkor a böngésző néha egy ahhoz hasonló üzenetet jelenít meg, hogy „Ön most az oldalt egy biztonságos kapcsolaton keresztül tekinti meg…". Az egyes böngészők más-más módon jelezhetik a biztonságos kapcsolatokat.
4. Miután a kapcsolat létrejött, általában valamilyen hitelesítési eljárásra kerül sor. A legtöbb tranzakciót bonyolító webhelyen a vásárlóknak valamilyen fiókot kell nyitniuk az eladónál. Ennek részben biztonsági okai vannak, részben pedig a kényelmet szolgálja (hogy a felhasználó nyomon követhesse a rendelései állapotát). A felhasználói fiók adatai azt is lehetővé teszik az eladónak, hogy figyelje a felhasználó tevékenységét, és megállapítsa a földrajzi elhelyezkedését,

valamint hogy milyen termékeket vásárolt korábban. A bejelentkezés során a webkiszolgálónak kapcsolatba kell lépnie egy háttérben működő adatbázis-kiszolgálóval – vagy azért, hogy új fiókot nyisson, vagy hogy ellenőrizze a meglevő fiókhoz tartozó azonosító adatokat.

5. Miután a felhasználó bejelentkezett, a kiszolgálónak (vagy egy másik alkalmazásnak, amely a háttérkiszolgálón működik) ellenőriznie kell a hitelkártya-adatokat, és jeleznie kell a tranzakciót egy hitelkártya-hatóságnak. Ez a hitelkártya-hatóság gyakran a hitelkártyát kibocsátó cég egy kereskedelmi szolgáltatása.
6. Ha a tranzakciót jóváhagyták, a rendelési és postázási információk a teljesítési osztályhoz kerülnek, a tranzakciókezelő alkalmazás pedig gondoskodik a rendelés megerősítéséről a felhasználó számára, illetve a felhasználó fiókprofiljának frissítéséről.

Az olyan operációsrendszer-gyártók, mint a Sun vagy a Microsoft, tranzakciós kiszolgálóalkalmazásokat nyújtanak, amelyek segítenek a webes rendelések feldolgozásában. Mivel a webes tranzakciók különlegesek, és a kereskedő hálózatán meglevő alkalmazásokkal kell együttműködniük, az alkalmazás-keretrendszerek általában különleges eszközökkel segítik a tranzakciós rendszer felépítését.



20.3. ábra

A webes tranzakciók jellemző forgatókönyve

Vegyük észre, hogy a 20.3. ábráról hiányzik a tűzfal szerepe a tranzakciós rendszerben. Egy nagy méretű kereskedelmi hálózatban általában tűzfal működik a webkiszolgáló mögött, amely a hálózatot védi, és egy a webkiszolgáló előtt is, amely bizonyos forgalmat blokkol, de a webes kérelmek számára nyitva hagyja a kiszolgálót. A nagy forgalmú webhelyeken ezen kívül egy helyett valószínűleg több webkiszolgálót találunk, amelyek megosztják a terhelést egymás között.

A webkiszolgáló és a háttérkiszolgálók között egy védett belső hálózat biztosíthatja a kapcsolatot, de a fő hálózattól független kijelölt vonal is vezethet a háttérrendszerhez. A hitelkártya-hitelesítő kiszolgáló általában egy a webhelyen kívüli szolgáltatás, amelyet egy másik cég nyújt, és az elérése biztonságos internetkapcsolaton keresztül történik.

Összefoglalás

A Web eszközei sokféle alkalmazás fejlesztését lehetővé teszik. Az egyszerű weboldalakon és webes űrlapokon kívül a fejlesztők összetett alkalmazásokat is építhetnek, amelyek képesek foglalásokat rögzíteni, raktárkészletet ellenőrizni, illetve rendeléseket feldolgozni. Ebben az órában a webszolgáltatásokkal kapcsolatos technológiák közül tekintettünk át néhányat. Megismertük a webszolgáltatási infrastruktúrát, és megtanultuk, mi a jelentősége. Ezen kívül a webszolgáltatások három fontos összetevőjéről, az XML-ről, az SOAP-ről és a WSDL-ről beszéltünk, végül pedig az órát a web alapú tranzakciók szerkezetének vizsgálatával zártuk.

Kérdezz-felelek

K *Mi az előnye a webszolgáltatási modellnek a hagyományos ügyfél-kiszolgáló rendszerű programozással szemben?*

V A webszolgáltatási modell olyan szabványos összetevőket – például webkiszolgáló és webböngésző alkalmazásokat – egyesít, amelyek már jelen vannak a legtöbb hálózaton.

K *Miért alapul a webszolgáltatási modell az XML-en és nem a HTML-en?*

V A HTML előre meghatározott elemek gyűjteménye, amelyeket kifejezetten weboldalak leírónyelvének szántak. Az XML ezzel szemben szinte korlátlanul alkalmas új elemek meghatározására és értékek változókhoz rendelésére.

K *Figyelembe véve, hogy számtalan gyártó saját nyelveket és összetevőket használ a webszolgáltatások támogatására, mi az előnye az olyan egységes szabványoknak, mint az SOAP vagy a WSDL?*

V Az olyan szabványok, mint az SOAP és a WSDL, közös formátumot biztosítanak, hogy a különböző gyártói környezetekhez írt összetevők képesek legyenek könnyen együttműködni.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **LAMP** – Nyílt forrású webszolgáltatási verem, amely a Linux operációs rendszerből, az Apache webkiszolgálóból, a MySQL adatbázis-rendszerből és a három „P" betűvel kezdődő nevű programozási nyelv (PHP, Perl és Python) valamelyikéből áll.
- **SOAP** – Üzenetcsere-protokoll webalkalmazások számára.
- **Webszolgáltatási architektúra** – A webes összetevőkre épülő egyéni hálózati alkalmazások fejlesztését szolgáló rendszer.
- **WSDL (Web Services Description Language, webszolgáltatás-leíró nyelv)** – XML alapú formátum hálózati szolgáltatások leírására.
- **XML (eXtensible Markup Language, bővíthető jelölőnyelv)** – Jelölőnyelv programadatok meghatározására és átvitelére a webszolgáltatásként működő alkalmazásokban.

21. ÓRA

Az új Web

A fejezet tartalmából:

- Újdonságok a Weben
- XHTML
- Fájlcserélő hálózatok
- IRC és IM
- A jelentésközpontú Web

Újdonságból temérdek van a Weben. Ezek új formákat és formátumokat szülnek, de ami kívülről újnak tűnik, az lehet, hogy csupán a meglevő eszközök és szolgáltatások ügyes összehangolása. Ebben az órában az új Web látképét tekintjük át.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni a webnaplók, a wiki-oldalak és a közösségi webhelyek működését.
- El tudjuk magyarázni az XHTML szerepét.
- Érteni fogjuk a fájlcserélő hálózatok működését.
- Le tudjuk írni az IRC és az IM üzenetküldő rendszerek működését.
- El tudjuk magyarázni a jelentésközpontú Web célját.

Web 2.0

A Világháló (World Wide Web) kinézete az utóbbi időben átalakult – okosabb, interaktívabb webhelyek új nemzedéke jelent meg, amelyek a felhasználók új generációját szolgálják ki. Az új webhelyeket működtető technológiákat hívják összefoglaló néven *Web 2.0-nak*.

Az új Web elemeinek megjelenése eltér a régi stílusú webhelyekétől, és az emberi interakció, valamint a közösségi élmény szempontjából forradalmi változást jelentenek, ugyanakkor a színfalak mögött a Web 2.0 technológiái logikusan következnek a webszolgáltatási infrastruktúra hasonló összetevőiből:

- **Adatbázis-rendszerek** – Az adatok tárolására és kezelésére szolgáló önálló rendszerek.
- **Tervezőelemek** – Készen kapott szabványos elemek.
- **Elrendezés** – A webhelyek szerkezete.
- **Parancskódok** – HTML-kód előállítása adatok befecskendezésével az előre kialakított szerkezetbe.

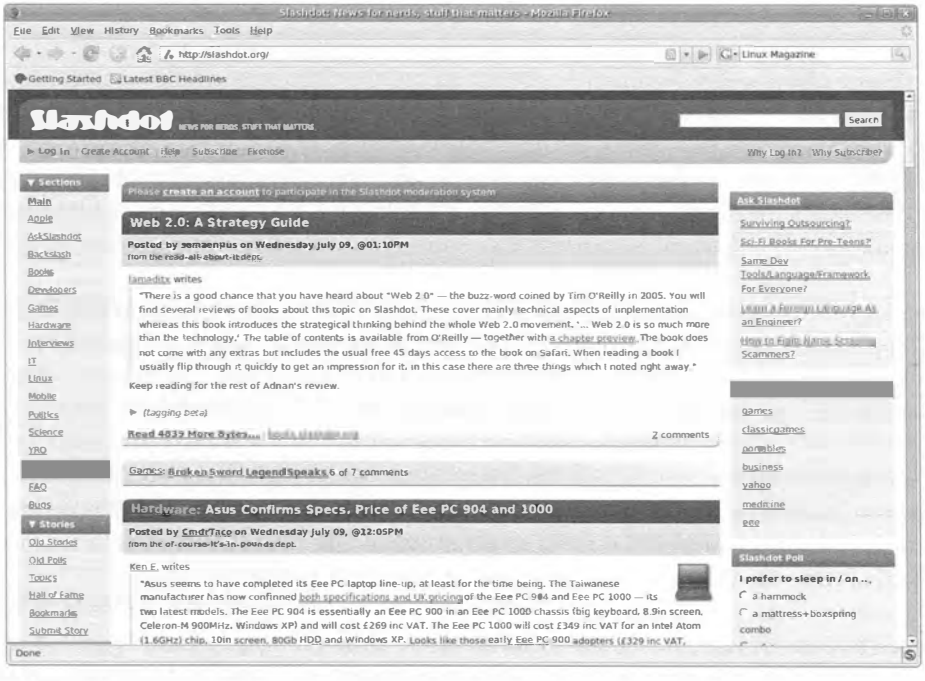
A Web olyan csodái, mint a webnaplók (blogok), a wiki-oldalak és a közösségi webhelyek, elrejtik ezeket a részleteket, így a felhasználó szabadon formálhatja a webes személyiségét képek, hangok és írott szöveg segítségével, anélkül, hogy valaha foglalkoznia kellene olyan bosszantó apróságokkal, mint a HTML.

Az új eszközök többségének egyik fontos közös jellemzője a *WYSIWYG* szerkesztőfelület. Ez a betűszó a What You See Is What You Get („azt kapod, amit látsz") rövidítése. Más szavakkal, a szöveget, a képeket és más elemeket olyan környezetben szerkeszthetjük, amely megegyezik azzal, amit a felhasználó látni fog. Ez az elgondolás nem is olyan radikális, mint hinnénk – lényegében így működnek a szövegszerkesztő programok, és az olyan webfejlesztő eszközök is, mint a Dreamweaver, már évek óta kínálnak ilyen szolgáltatást. Az új webes eszközök esetében azonban a WYSIWYG szerkesztőfelület elmossa a határt a webes felhasználó és a webfejlesztő között, és olyan környezetet alakít ki, amelyben a felhasználó egyszerre öltheti magára a fogyasztó és a webtartalom-készítő szerepét.

A következőkben az új Web néhány fontos elemét tekintjük át. Ahogy a leírásokat olvassuk, képzeljük magunk elé, hogy mi történhet valójában a képernyő egyszerű, tiszta nézete mögött: a webügyfél kapcsolódik egy webkiszolgálóhoz, a webkiszolgáló pedig dinamikus HTML-t ad át neki, az alapelrendezés keretét az oldal XML alapú adataival feltöltve, amikor pedig a felhasználó megváltoztat valamit az oldalon, az ügyfélböngésző frissítést küld az oldalhoz tartozó adatokat tároló adatbázisnak.

Webnaplók

A *webnapló* vagy *blog* (ez a *weblog*, vagyis webnapló rövidítése) egy „e-zine" vagy elektronikus újság, amelynek a tetején az új hírek vagy történetek találhatók, a régebbieket pedig az oldalt lefelé görgetve érhetjük el. A blog időrendi jellege azt a benyomást kelti, hogy az oldal folyamatosan változik és átalakul, ami visszatérésre készteti az olvasókat. Egyes webnaplóírók („bloggerek") lényegében internetes naplót írnak, de a formátumot kritikusok, tudósítók és vállalati szóvivők is használják. Sok blog, például a számítástechnika híreire és újdonságaira kíváncsi olvasók kedvence, a Slashdot.org, valójában hírportál.



21.1. ábra

A Slashdot.org egy népszerű webnapló

A webnaplókat általában a webkiszolgálón futó webnaplóíró szoftverrel készítik. A Slashdothoz a Slash nevű eszközt használják, amely egy nyílt forrású alkalmazás, és ingyenesen letölthető a SourceForge webhelyéről, a `http://sourceforge.net/projects/slashcode/` címről. Más webnaplóíró alkalmazások, például a WordPress, illetve különféle, a blogokat támogató tartalomkezelő rendszerek szintén szabadon hozzáférhetők. A Microsoft asztali webnaplóíró programként a Windows Live Writert kínálja.

A webnaplók működésének vizsgálatára az egyik módszer az ügyfélnek küldött forráskód megtekintése. A legtöbb webböngésző lehetővé teszi a webes dokumentumok forráskódjának megtekintését. A Slashdot esetében azt láthatjuk, hogy a különféle új bejegyzéseket egymásba ágyazott `<div>` HTML-elemekkel hozzák létre. A `<div>` címke egy szakaszt jelöl a dokumentumon belül. A böngészőben megtekinthető kód a kész HTML, amelyet az ügyfél kap. A kiszolgálóoldalon ezt a kódot egy alkalmazás vagy parancsfájl (a Slashdot esetében a Slash alkalmazás) állítja elő, jellemzőértékeket beszúrva az olyan elemek számára, mint a title (cím), a description (leírás), az introduction (bevezetés), az image (kép), és így tovább. Ezek az elemek a bejegyzéshez tartozó adatrekordból származnak.

Wiki-oldalak

A *wiki* egy olyan webhely, amely közös szerkesztésre és információmegosztásra szolgál. A wiki-oldalak célja, hogy helyet biztosítsanak a felhasználóknak, ahol feljegyzéseket, dokumentumokat és más fontos információkat tehetnek közzé. Ideális esetben egy wiki könnyen bővíthető; a felhasználók egyszerűen hozhatnak létre új oldalakat, és kapcsolhatják azokat a meglevő oldalakhoz. Egyes wiki-oldalak változatkövetést is biztosítanak, így a különböző felhasználók módosításai nyomon követhetők.

A legnagyobb wiki a világon a Wikipedia nevű hatalmas enciklopédia (lásd a 21.2. ábrát). A Wikipedia felhasználói saját szócikkeket írhatnak, és szerkeszthetik a meglevőket. (Ha látni szeretnénk egy bejegyzés módosításait, kattintsunk a Recent Changes – Friss változtatások – hivatkozásra a Wikipedia menüjében.)

Egyes cégek és szervezetek kiterjedten használnak wiki-oldalakat tervezési célokra, a munka összehangolására, illetve a dokumentumok rendszerezésére. A Wikipedia webhelyen használt MediaWiki nevű szoftver ugyancsak nyílt forrású, szabadon hozzáférhető alkalmazás (`http://www.mediawiki.org/wiki/MediaWiki`).

A wiki-rendszerek felépítése különböző lehet, de abban megegyeznek, hogy a wiki-oldalak vagy -bejegyzések (például a Wikipedia szócikkei) szabványos jellemzőkhöz rendelt értékek gyűjteményei. Az egyes bejegyzésekhez tartozó értékeket egy XML-séma vagy más hasonló adatszerkezet határozhatja meg. Az értékek ehhez hasonlók lehetnek:

- **Title (Cím)** – A bejegyzés címe.
- **Category (Kategória)** – A bejegyzés témakör szerinti besorolása.
- **Language (Nyelv)** – A bejegyzés nyelve.
- **Contents (Tartalom)** – A bejegyzés teljes HTML-kódja.

21.2. ábra

A Wikipedia egy hatalmas wiki-oldal, amelyet bárki szerkeszthet

Ennek a szerkezetnek a bővítésein keresztül a szöveg módosításait is nyomon lehet követni. Amikor az oldalt lekérik, az adatok összeolvadnak az elrendezést kialakító elemekkel és más formázási információkkal, és létrejön belőlük az a kód, amelyet a böngésző feldolgoz és megjelenít.

Közösségi webhelyek

A Facebook, a MySpace és más *közösségi webhelyek* ma már a kultúránk részét képezik. Ezeket a szolgáltatásokat arra tervezték, hogy a felhasználók személyes weboldalakat építhessenek bármiféle HTML-ismeret nélkül. Sok közösségi webhely olyan szolgáltatásokat is nyújt, mint a webnaplóírás lehetősége, illetve az azonnali üzenetküldés, és a legtöbbjük lehetőséget ad képek feltöltésére, illetve arra, hogy zenét társítsunk a napjainkról szóló beszámolókhoz.

Az elv ugyanaz, mint más szolgáltatások esetében: a felhasználóhoz különféle jellemzők tartoznak, amelyeket egy adatbázis tárol, és amikor az oldalt lekérik, a kiszolgálón futó szoftver összeolvasztja a felhasználóhoz tartozó adatokat a webhely szerkezetét leíró általános sablonnal, hogy létrehozza a böngészőben megjelenő nézetet.

XHTML

Az új Web számos eszköze, illetve az Interneten ma található sok-sok webhely egy olyan újdonságra támaszkodik, ami ennek az órának a témájához képest meglehetősen műszaki jellegű, de mindenképpen megéri szót ejteni róla. Az *XHTML* szabványról van szó, amelynek célja, hogy áthidalja a régimódi HTML és az XML alapú webes környezet valósága közötti szakadékot. Az XHTML lényegében a HTML szolgáltatásainak olyan megfogalmazása, amely megfelel az XML nyelvtanának. Az XHTML formátum a HTML teljes kifejezőkészségét biztosítja a gép által olvasható XML-sémák keretein belül.

Bár az XHTML-ben ugyanazokat az elemeket használjuk, mint a HTML-ben, az XHTML sokkal kényesebb a hanyag vagy nem szabványos kódolásra. Egyes elemeket másképp kell bevezetni, vagy formálisabb módon kell meghatározni, az elemek egymásba ágyazásának pedig pontosabbnak és szigorúan megszerkesztettnek kell lennie. A HTML XML-sémaként történő leírásának az a célja, hogy a fejlesztők rugalmasan készíthessék el azokat a parancsfájlokat és más programokat, amelyek előállítják és értelmezik a kódot. Az XHTML emellett alkalmasabb a dinamikus értelmezésre, illetve módosításra a fogadó fél által. Egy mobileszköz apró képernyője például nem biztos, hogy képes egy HTML-oldalt az előírt módon megjeleníteni, de egy az oldalt XHTML-ként fogadó ügyfélalkalmazás a kisebb képernyőhöz igazíthatja a szöveget.

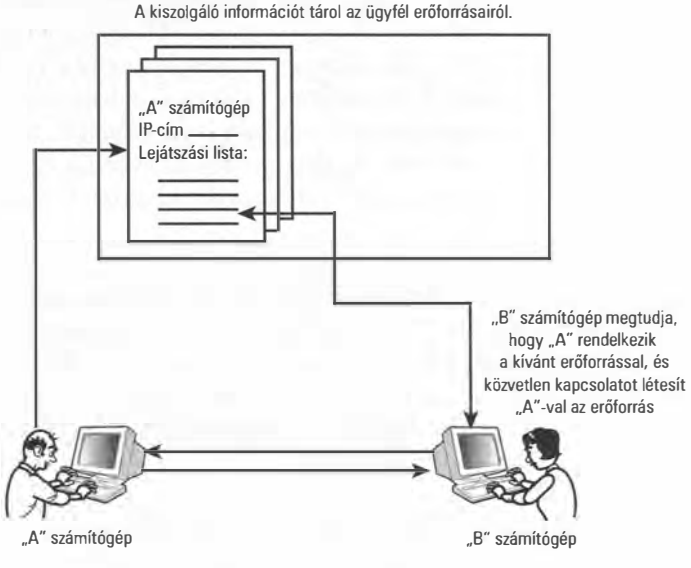
Fájlcserélő hálózatok

Az olyan internetes zenemegosztó közösségek, mint a Napster, egy új információmegosztási módszert hívtak életre: a *fájlcserélő* (peer-to-peer, röviden *P2P*, vagyis egyenrangú) *hálózatokat*. Magát a „peer-to-peer" vagy „egyenrangú" kifejezést a helyi (LAN) hálózatok hasonló kiépítéséből kölcsönözték, amelyben a szolgáltatások decentralizáltak, és minden számítógép egyaránt működik ügyfélként és kiszolgálóként. Az Interneten a P2P lehetővé teszi a hálózaton található számítógépeknek, hogy adatmegosztó közösségekben adatokat osszanak meg egymással. Más szavakkal, az adatok nem egyetlen webkiszolgálóról érkeznek, amely ügyfelek tömegét szolgálja ki, hanem a közösség tagjainak közönséges PC-in tárolódnak.

Ha figyelmesen olvastuk ezt a könyvet, bizonyára felmerül bennünk a kérdés, hogy miben különbözik a fent vázolt egyenrangú hálózat a közönséges hálózatoktól. Nos, az előző bekezdésben csupán annyit mondtunk, hogy a hálózat minden tagjának képesnek kell lennie (adatokat kérő) ügyfélként és (a kéréseket teljesítő) kiszolgálóként is működnie. A válasz tehát röviden az, hogy miután a kapcsolat létrejött, a P2P hálózat *valóban* közönséges egyenrangú hálózat. A hosszabb válasz magyarázza meg, miért tekintik többé-kevésbé forradalminak a P2P hálózatokat.

Az Internet létrehozásakor a sokszínűség volt az egyik cél, és elméletileg minden internetkész számítógép létesíthet kapcsolatot bármely más kompatibilis internetkész számítógéppel, amely rendelkezik a kívánt szolgáltatásokkal. Figyelembe kell vennünk azonban, hogy a közönséges PC-k nincsenek mindig bekapcsolva, illetve hogy az Internetre kapcsolódó legtöbb számítógép nem rendelkezik állandó IP-címmel, hanem dinamikus címet kap egy DHCP-kiszolgálótól (lásd a 12. óra anyagát). Egy hagyományos TCP/IP-hálózaton a számítógépek nem tudhatják, hogyan léphetnek kapcsolatba egy másik géppel, ha annak nincs állandó IP-címe vagy tartományneve.

A fájlcserélő hálózatok tervezői tudták, hogy egy sokszínű zenemegosztó közösség nem működhet, ha nem oldják meg ezeket a problémákat. A megoldásuk az volt, hogy a kapcsolati információkat egy központi kiszolgáló szolgáltassa, amelynek a segítségével aztán az ügyfelek kapcsolatot létesíthetnek egymással. Nézzük meg a 21.3. ábrát: az „A" számítógépnél ülő felhasználó kapcsolódik az Internetre, a PC-jén működő ügyfélszoftver pedig bejelenti a felhasználó jelenlétét a kiszolgálónak, amely tárolja az ügyfél IP-címét, illetve feljegyzi azokat a fájlokat, amelyeket az ügyfél elérhetővé tesz a közösség számára. A „B" számítógépnél ülő felhasználó kapcsolódik a kiszolgálóhoz, és felfedezi, hogy az egyik fájl, amelyre szüksége lenne, az „A" számítógépen található. A kiszolgáló átadja a „B" számítógépnek az „A" számítógép eléréséhez szükséges információkat, „B" pedig közvetlen kapcsolatot létesít „A"-val, és letölti a fájlt.



21.3. ábra

A fájlcserélő számítógépek bejegyeztetik a címüket és az erőforrásaik listáját, amelyeket aztán más számítógépek közvetlen kapcsolaton keresztül érhetnek el

A fájlcserélő közösségek legnagyobb előnye, hogy az IP-cím kérésének és a kapcsolat létrehozásának részleteit a szoftver kezeli. A felhasználó a fájlcserélő alkalmazás felhasználói felületének keretein belül marad, tehát semmit nem kell tudnia a hálózat működéséről.

A fájlcserélő hálózatokat sokan támadják, de nem a technológia miatt – a probléma tisztán jogi természetű. A fájlcserélő hálózatok kifejlesztésének egyik célja ugyanis a jogvédett anyagok lenyomozhatatlan (és néha illegális) cseréje volt.

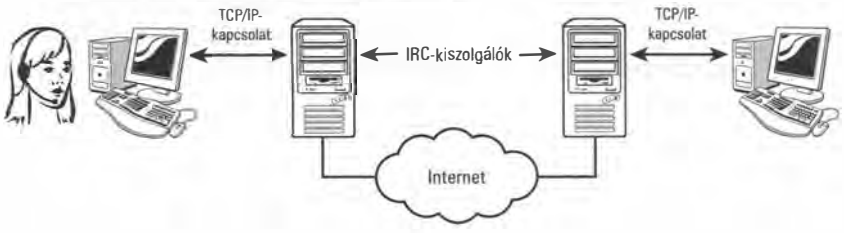
IRC és IM

Már évek óta léteznek valósidejű szöveges üzenetküldő rendszerek. Valójában az elv még az Internet születésénél is korábbra nyúlik vissza, de az utóbbi időben az internetes csevegés (chat) nagy népszerűségre tett szert a felhasználók új nemzedékének körében. Sok felhasználó szívesebben él az azonnali üzenetküldés lehetőségével, mint hogy felemelje a telefonkagylót. Csevegni ugyanis a számítógépen végzett munka közben is lehet, és az írott szöveg kevésbé tolakodó, mint a hang, így alkalmasabb arra, hogy párhuzamosan több dolgot csináljunk egyszerre. Egyesek még arra is képesek, hogy egyidejűleg több csevegést folytassanak – ami telefonon sokkal nehezebb.

Az üzenetküldésnek ma számos formája létezik – vannak jogvédett és nyílt rendszerek is. A népszerű *IRC*-t (Internet Relay Chat, *internetes csevegés*) egy sor RFC-dokumentum (RFC 1459, majd RFC 2810–2813) írja le. Az IRC valójában egy protokoll, amely a TCP/IP alkalmazásrétegében működik. Az IRC protokollt hivatalosan a 194-es TCP-kapuhoz rendelték, de a kiszolgálók jellemzően a magasabb sorszámú kapukat használják, hogy elkerüljék a rendszergazdai jogokkal való működést. Az IRC-hálózatok olyan IRC-kiszolgálókból állnak, amelyek egymással kommunikálva támogatják a hálózat felhasználóinak interaktív csevegését.

A csevegési munkamenet egy úgynevezett IRC-csatornán zajlik. A csatornához a hálózat tetszőleges helyéről több felhasználó is csatlakozhat, és valód időben társaloghatnak egymással (lásd a 21.4. ábrát). A csatornákon közös szakmai vagy személyes érdeklődés, illetve családi vagy baráti kapcsolatok szerint csoportok alakulhatnak. Elméletben van egy kijelölt felhasználó (a csatornakezelő, „channel op"), aki a csatornáért felel, és jogában áll felhasználókat kitiltani, vagy moderálni a tartalmat.

A felhasználóknak egy ügyfélprogram teszi lehetővé, hogy kapcsolódjanak egy IRC-kiszolgálóhoz, és csatlakozzanak egy csevegőcsatornához. Szöveges csevegés esetén az ügyfelek szöveges parancsokkal kommunikálnak egymással. Az újabb, grafikus felületű eszközöknek köszönhetően a parancsok nyelvtanát már nem fontos ismerni; a felhasználó immár pontosan úgy írhatja be a szöveget, mintha személyesen beszélgetne.



21.4. ábra

Az IRC-kiszolgálók fogadják a felhasználóktól érkező kapcsolatokat, és a hálózat más kiszolgálóival kommunikálnak

A világon sok-sok IRC-hálózat létezik. A legnagyobbak, mint az EFnet, állítólag akár 30 000 felhasználónak tesznek lehetővé egyidejű csevegést. Az IRC-hálózatokhoz könnyű csatlakozni. A feliratkozáshoz és a bejelentkezéshez néha csupán egy internetes becenév (nickname) szükséges. Egyesek megpróbálnak szigorúbb biztonsági intézkedéseket életbe léptetni, de az IRC-t soha nem tervezték különösebben biztonságosnak.

Az *azonnali üzenetküldés* (Instant Messaging, *IM*) elve hasonló a csevegéséhez, de kevésbé szabványosított, és általában több lehetőséget kínál. A felhasználónak jellemzően csak fel kell iratkoznia, letöltenie egy ügyfélprogramot, és kicserélnie az eléréséhez szükséges adatait azokkal a barátaival, akik szintén csatlakoztak a hálózathoz. Az azonnali üzenetküldő rendszerek többnyire jogvédettek, és a hálózatokat nagy internetes cégek üzemeltetik. A legnagyobb azonnali üzenetküldő hálózat az AOL Instant Messenger (AIM) hálózata, 50 millióra becsült aktív felhasználóval. Az egyéb magántulajdonú rendszerek közül említésre méltó még a Windows Live Messenger Network és a Yahoo Network.

Az egyik népszerű nyílt forrású azonnali üzenetküldő az *XMPP*-n (eXtensible Messaging and Presence Protocol, bővíthető üzenetküldési és jelenléti protokoll) alapul, amelyet a Jabber hálózaton használnak. Az XMPP egy XML alapú protokoll csevegőüzenetek cseréjéhez. A Jabber hálózatnak mintegy 40 millió felhasználója van szerte a világon.

A jelentésközpontú Web

Egy ígéretes kutatási terület, amely valóban újabb forradalmat idézhet elő az Interneten, a *jelentésközpontú* (szemantikus) *web* nagyratörő terve. A jelentésközpontú Web, amelyet a World Wide Web megalkotója, Tim Berners-Lee lelkesen támogat és népszerűsít, a tervek szerint egy általános módszer arra, hogy a webes adatokhoz valódi, emberi jelentést kapcsoljanak. Más szavakkal, a cél a webes információk jelentésének olyan kódolása, amelyet egy számítógép könnyen el tud érni és fel tud dolgozni.

A jelentésközpontú Web céljának megértéséhez először is tisztában kell lennünk vele, hogy egy weboldalon valójában milyen kevés „tudás" van jelen. Vegyük például az alábbi szövegsorokat, amilyenekkel bármely közönséges webhelyen találkozhatunk:

A vágy villamosa
Lawrence Community Theater
Szombat, október 12., 2008
7:30 PM

Egy ember erre a szövegre pillantva rögtön tudja, hogy egy olyan eseményt reklámoz, amelyre a Lawrence Community Theaterben kerül sor október 12-én 7 óra 30 perckor. Sokan *A vágy villamosáról* is tudják, hogy egy híres színdarab címe, de a *Theater* (színház vagy filmszínház) szóból még azok is kikövetkeztethetik, hogy egy filmről vagy egy színműről van szó, akik soha nem hallottak róla.

Ezzel szemben egy számítógép csak alfanumerikus szöveget lát, a jelentéséről semmit sem tud. Nem tudja, mi az a „theater", sőt azt sem, hogy a harmadik sor egy időpont, hacsak kifejezetten meg nem mondjuk neki. Egy keresőprogram akár arra is találatként adhatja ezt az oldalt, ha egy felhasználó a villamosok menetrendjére kíváncsi.

A jelentésközpontú Web eszközei egy napon lehetővé teszik majd, hogy a webfejlesztők kódolják a jelentésre vonatkozó információkat (szemantikai információkat), hogy az automatizált folyamatok felismerjék, hogy a fenti szöveg egy színdarabra, és nem a villamos-menetrendre vonatkozik. Mivel a szemantikai információk magába az oldalba lesznek kódolva, a webhely készítőjének nem kell semmilyen előzetes ismerettel rendelkeznie arról, hogy az olvasók miként fogják majd felhasználni az információt. Bárki készíthet később egy eszközt, amely színdarabokról keres információt, és az eszköz rá fog bukkanni erre a színházi hirdetésre. A különböző webhelyek más-más módon jeleníthetik meg az információkat, szabványos formátum vagy stílus nélkül, a színdarabkereső alkalmazás akkor is megtalálja az előadásokat – amíg a szemantikai információ meghatározza a szöveg jelentését.

A jelentésközpontú webes eljárások még kísérleti stádiumban vannak, bár néhányat már közzétettek a W3C (World Wide Web Consortium) kiadványaiban. Az egyik jelentésközpontú webes eszköz, amely jelentős figyelmet kapott a webes közösségben, az RDF (Resource Description Framework, *erőforrás-leíró keretrendszer*) névre hallgat. Az RDF olyan keretrendszer, amely a jelentésre utaló kapcsolatok kifejezésére szolgál. Az RDF alapegységei az utasítások, amelyek három részből állnak – ezt hívják az RDF nyelvezetében „triple"-nak. A „triple" vagy „hármas" úgy épül fel, mint egy egyszerű mondat, amelynek van egy alanya (subject), egy állítmánya (predicate) és egy tárgya (object).

Abban a mondatban például, hogy „The play has the title A Streetcar Named Desire" (A színdarab címe *A vágy villamosa*) az alany „The play", a tárgy „*A Streetcar Named Desire*", az állítmány pedig a „has the title".

Az RDF-hármasok többféle alakot ölthetnek, de az alapelv az, hogy minden elemet egy URI-ként fejezünk ki, és egy kettőspontokkal elválasztott listában fűzzük össze őket. A Dublin Core Metadata Initiative egy adatbázisban gyűjti össze az RDF-hármasokban hivatkozott szabványos állítmányokat. A `<http://purl.org/dc/elements/1.1/title>` kód például a „has the title" állítmányra hivatkozik.

Az RDF és más jelentésközpontú webes technológiák egy napon talán intelligensebb keresőeszközök készítését teszik lehetővé.

Összefoglalás

Ebben az órában az új Web néhány eszközével és technológiájával ismerkedtünk meg. Megtanultuk, hogy a webnaplók, a wiki-oldalak és a közösségi webhelyek olyan összetevőkre támaszkodnak, mint az adatbázisok vagy a dinamikus HTML, és láttuk, hogyan biztosítja a HTML lehetőségeit az XHTML egy egységes XML-sémán keresztül. Az órán ezenkívül beszéltünk a fájlcserélő hálózatokról, az üzenetküldő szolgáltatásokról és a jelentésközpontú Webről is.

Kérdezz-felelek

K *Miért érdemes wiki-oldalakat használni a hagyományos webhelyek helyett?*

V A wiki-oldalak könnyen bővíthetők és módosíthatók, és úgy alkották meg őket, hogy támogassák a közös munkát. Sok wiki-oldal beépített változatkövető rendszerrel rendelkezik, amely nyomon követi a különböző felhasználók módosításait.

K *Mi a legjellemzőbb tulajdonsága a fájlcserélő hálózatoknak?*

V Minden csomópont képes ügyfélként és kiszolgálóként is működni.

K *Mi az előnye a jelentésközpontú kódolásnak a Weben?*

V A jelentésleíró információk az adatok kifinomultabb értelmezését teszik lehetővé a keresőprogramok és más eszközök számára.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Blog (weblog, webnapló)** – Rendszeresen frissített, híreket vagy történeteket üzenetek függőlegesen, időrendben elrendezett sorában közzétevő webhely.
- **IM (Instant Messaging, azonnali üzenetküldés)** – Valósidejű üzenetküldési módszer.
- **IRC (Internet Relay Chat, internetes csevegőszolgáltatás)** – Protokoll és hálózati szolgáltatás valósidejű szöveges üzenetküldéshez.
- **P2P (Peer-to-Peer) vagy fájlcserélő hálózat** – Internetfelhasználók között fájlok megosztása céljából közvetlen kapcsolat létesítésére szolgáló rendszer.

- **RDF (Resource Description Framework, erőforrás-leíró keretrendszer)** – Jelentésközpontú webes keretrendszer.
- **Jelentésközpontú Web** – Azoknak a technológiáknak az összessége, amelyeknek a célja, hogy információt nyújtsanak a webes adatok jelentéséről.
- **Közösségi webhely** – Szolgáltatás, amely személyes weboldalakon keresztül támogatja a webnaplóírást, az üzenetküldést és más tevékenységeket.
- **Web 2.0** – Az interaktív Web új vízióját tükröző eszközök összessége.
- **Wiki** – Egyszerűen szerkeszthető, interaktív webhely, amelyet a felhasználók közösen építhetnek fel.
- **WYSIWYG (What You See Is What You Get, „azt kapod, amit látsz")** – A szerkesztőeszközöknek az a típusa, amely az oldalakat úgy jeleníti meg, ahogy azokat a felhasználó látni fogja.
- **XHTML** – XML-sémán keresztül kifejezett HTML.
- **XMPP (eXtensible Messaging and Presence Protocol, bővíthető üzenetküldési és jelenléti protokoll)** – Nyílt forrású üzenetküldési protokoll, amelyet a Jabber üzenetküldővel használnak.

22. ÓRA

Hálózati támadások

A fejezet tartalmából:

- A támadók típusai
- A támadók céljai
- Támadási módszerek

Amikor a szakemberek hálózatokat kezdtek tervezni, fogalmuk sem volt arról, hogy jogosulatlan felhasználók serege fog több ezer órát tölteni azzal, hogy megpróbál behatolni ezekbe a hálózatokba. Ezen az órán a hálózati támadásokhoz alkalmazott módszerek közül ismerkedünk meg néhánnyal.

Az óra végeztével a következőkre leszünk képesek:

- Le tudjuk írni, hogyan szerzik meg a támadók a jelszavakat.
- El tudjuk magyarázni, hogyan férnek hozzá a támadók a rendszerekhez olyan módszerekkel, mint az átmeneti tárak túlcsordulásának előidézése vagy a munkamenetek eltérítése.
- Le tudjuk írni az adathalászat működését.
- El tudjuk magyarázni, hogyan működik a szolgáltatás-megtagadást előidéző támadás.

Vandálok és kiberbűnözők

Az Internet növekedése korlátlan lehetőséget teremtett a behatolók számára, hogy titkokat lopjanak el, webhelyekbe piszkáljanak bele, hitelkártya-információkkal éljenek vissza, vagy egyszerűen csínytevéseket kövessenek el. Az internetes támadók ezen kívül új mítoszokat hívtak életre: tudásukat és merészségüket sokan csodálják – egyesek még magasröptű művészi vagy politikai célokat is tulajdonítanak ezeknek a szélessávú útonállóknak. A számítógép-hálózatokat kiépítő és karbantartó szakembereket azonban egyáltalán nem nyűgözi le a hálózati támadók tevékenysége.

Ezen az órán azok közül a módszerek közül ismerkedünk meg néhánnyal, amelyeknek a segítségével a támadók megkaparintják a számítógép-rendszerek feletti vezérlést. Miközben ezeket a módszereket tanulmányozzuk, észre fogjuk venni, hogy a mögöttük álló elgondolások a TCP/IP-nek azokra az alapvető tulajdonságaira támaszkodnak, amelyeket az előző órák során megismertünk. Azokkal a biztonsági intézkedésekkel, amelyek azt a célt szolgálják, hogy a támadókat távol tartsuk a hálózattól, a következő órán foglalkozunk.

Az Internettel kapcsolatos szakirodalom tömve van homályos pszichológiai profilokkal, amelyek megpróbálják leírni, hogy kik is ezek a támadók, és hogyan gondolkodnak. Az információk többsége azonban anekdotákon és spekuláción alapul. Abban mindenesetre általános az egyetértés, hogy a számítógépes támadók többnyire az alábbi csoportokból kerülnek ki:

- **Tinédzser amatőrök** – Ebbe a csoportba olyan gyerekek tartoznak, akik csupán játszanak. Az úgynevezett *szkriptkölykök* gyakran csak felületes tudással rendelkeznek a számítógéprendszerekről, és elsősorban az Interneten hozzáférhető behatolási parancsfájlokat és eljárásokat alkalmazzák.
- **Rekreációs céllal támadók** – A „felnőtt" támadóknak ebbe a csoportjába a legkülönfélébb motivációjú egyének tartoznak. A legtöbben tisztán szellemi kihívást keresnek, míg mások egy adott iparággal vagy szervezettel szemben szeretnének véleményt nyilvánítani, de akadnak köztük haragos korábbi alkalmazottak is.
- **Szakmabeliek** – Ez a veszélyes csoport olyan tapasztalt szakemberekből áll, akik alaposan ismerik a számítógépeket. Az ő nyomukat nehéz követni, mert minden trükköt ismernek – néhányat egyenesen ők dolgoztak ki. Ezeket a támadókat kizárólag az anyagi haszonszerzés ösztönzi, bár nem jutottak volna el arra a szintre, ahol vannak, ha nem szeretnék, amit csinálnak. A profik közül sokan olyan tevékenységekre összpontosítanak, mint a hitelkártya-csalás vagy a személyiséglopás. Az egyik újabb divat az otthoni számítógépek megtámadása abból a célból, hogy a rendszereiket levélszemét küldésére használják fel.

Lehetetlen az összes fogást leírni, amit a támadók arra használnak, hogy hozzáférést szerezzenek a számítógéprendszerekhez, ezért ezen az órán csak a legfontosabb módszereket tekintjük át röviden. Miközben az itt leírt eljárásokat böngésszük, vessük az eszünkbe a számítógépek biztonságának legfontosabb szabályát: ha azt hisszük, hogy a hálózatunkat megfelelően biztosítottuk, gondoljuk át még egyszer, mert valaki időt és munkát nem sajnálva most is biztosan azon igyekszik, hogy behatolási pontot találjon rajta.

Mit akarnak a támadók?

Ahogy az előző részben említettük, a hálózati támadókat a legkülönfélébb ösztönzők vezérlik. Mindazonáltal, bár a céljuk különböző lehet, közös bennük, hogy egy számítógéprendszer vagy hálózat feletti uralom megszerzésére törekszenek, ezért hasonló lépéseket hajtanak végre.

A számítógépes támadások és a beszivárgás folyamata a következő alapvető lépésekből áll:

1. A rendszerhez való hozzáférés megszerzése.
2. Jogosultságok szerzése.
3. Berendezkedés.
4. Felkészülés a következő támadásra.

Azt is érdemes megjegyezni, hogy a számítógép-hálózatok elleni összehangolt, jól szervezett támadások esetében ezeket a lépéseket gyakran előzi meg egy felderítési szakasz.

A támadók sokféleképpen juthatnak be a hálózatba, és rendezkedhetnek be ott. Bár az összes módszert lehetetlen lenne felsorolni, az eljárásokat három alapcsoportra oszthatjuk:

- **Azonosítók elleni támadások** – Ezek a támadások a rendszerbe történő normál bejelentkezéshez szükséges azonosítók megszerzésére összpontosítanak. A támadásra lényegében még az előtt kerül sor, hogy a támadó átszivárogna a biztonsági rendszeren. Az ilyen támadások egyik változata a *jogosultsági szint megemelése*, amikor is a támadó alacsony szintű hozzáférést szerez, majd különféle módszerekkel magasabb szintre tornássza fel magát.
- **Hálózatszintű támadások** – Az ilyen fajta támadások során a támadó általában egy nyitott kapun, nem biztosított szolgáltatáson vagy a tűzfal egy résén keresztül jut be a rendszerbe, de léteznek olyan hálózatszintű támadási módszerek is, amelyek a TPC/IP protokollrendszer kevésbé ismert tulajdonságait használják információszerzésre, illetve a kapcsolatok eltérítésére.

- **Alkalmazásszintű támadások** – Ennél a támadástípusnál a támadó a rendszeren futó egyik alkalmazás – például a webkiszolgáló – programkódjának ismert hibáit használja ki, hogy az alkalmazást bizonyos parancsok végrehajtására vagy más olyan viselkedésre vegye rá, ami a programozónak soha nem állt szándékában.

Egy „egész pályás" hálózati támadás sokszor a fenti módszerek kombinációjára épül. Szokványos forgatókönyv, hogy a támadó a kezdeti behatoláshoz valamilyen alkalmazásszintű eljárást alkalmaz, majd a jogosultságait rendszergazdai szintre emeli, és rejtett *hátsó ajtót* nyit, amelyen keresztül korlátlan hozzáférést tesz lehetővé a rendszerhez.

Egy másik hatékony támadási módszer, amely nem eredményez hozzáférést a rendszerhez, de ugyanakkor romboló hatású, az úgynevezett *elárasztásos vagy szolgáltatás-megtagadásos támadás* (denial-of-service), amelynek során a támadó a rendszer túlterhelését vagy összeomlását idézi elő, hogy az ne tudjon normálisan működni. Az elárasztásos támadásokról az óra később részében még részletesen beszélünk.

A vállalati hálózatok ellen intézett teljeskörű támadások egy széles felderítéssel kezdődnek, amelynek során a lehető legtöbb információt igyekeznek összegyűjteni a vállalatról. Ezt az eljárást néha lábnyomlevételnek (footprinting) is nevezik. Az információk egy része, például a vállalat székhelye, e-mail címei, fiókirodái vagy partnerwebhelyei, a Weben keresztül is beszerezhető. A támadó a vállalat által használt valamennyi tartománynevet igyekszik kideríteni, majd ezek segítségével lekérdezéseket intéz a DNS-kiszolgálókhoz, hogy megtudja a vállalat IP-címeit.

Azonosítók elleni támadások

A számítógép-rendszerekhez való hozzáférés megszerzésének klasszikus módja a bejelentkezéshez szükséges jelszó kitalálása. Ha a támadó interaktív hozzáférést szerez egy rendszerhez, akkor más eljárásokkal rendszerszintű jogosultságokhoz juthat, ezért egy hálózat feltörésének általában mindig egy jelszó – bármilyen jelszó – megszerzése az első lépése. A jelszavak megszerzésére különféle módszerek léteznek, a kifejezetten technikai jellegűektől (jelszófeltörő szótár-parancsfájlok és titkosítás-visszafejtő programok használata) a kifejezetten személyes fogásokig (kukák feltúrása, kutakodás a felhasználók asztalfiókjában). Az alábbi módszerek a jelszavak megszerzésére irányuló leggyakoribb támadások közé tartoznak:

- Számítógép nélküli információszerzés
- Trójai programok használata
- Találgatás
- Jelszóelfogás

A következőkben azt vizsgáljuk meg, hogyan lehet a fenti módszerek segítségével titokban megszerezni a felhasználók jelszavait.

Számítógép nélküli információszerzés

Nem számít, mennyire biztonságos a rendszerünk, a hálózatunk nem lesz biztonságos, ha a felhasználók nem védik a jelszavaikat. A jelszavak feltörésének egyik leggyakoribb oka a felhasználók óvatlansága. Az első számítógépes támadók gyakran úgy szereztek meg jelszavakat, hogy a számítógépről kinyomtatott, majd kidobott papírok között kerestek árulkodó információkat. Szerencsére azóta az operációs rendszerek gyártói kifinomultabban védik a jelszavakat, mégis rengeteg jelszót lopnak el számítógép nélküli információszerzés útján. A felhasználók elárulják a jelszavukat másoknak, vagy leírják azokat, és valamilyen könnyen hozzáférhető helyre teszik. A munkahelyek fizikai biztosítása általában kevésbé szigorú, mint a hálózaté: takarítók, elégedetlen munkatársak, sőt még jogosulatlan kívülállók is gyakran szabadon mászkálhatnak az irodákban, jelszóra utaló információkra vadászva. Amikor egy alkalmazott kilép, vagy elbocsátják, a fiókját általában törlik – de mi a helyzet azokkal a felhasználói fiókokkal, amelyek megosztott jelszót használtak a korábbi alkalmazottéval?

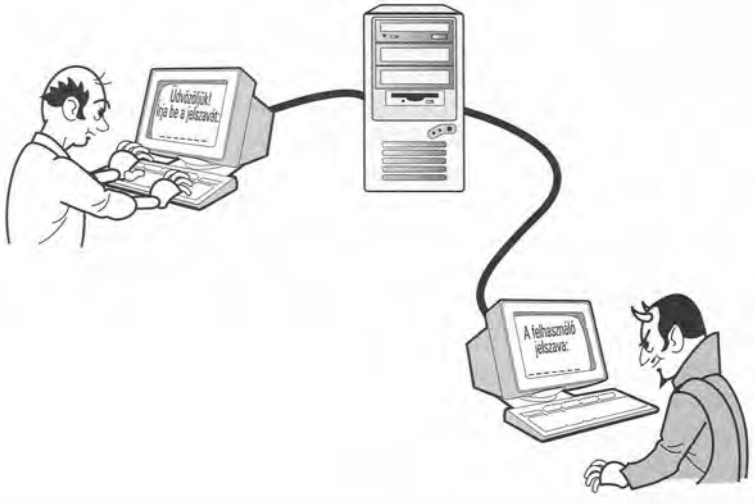
Egyes tapasztalt támadók jó képességekkel rendelkeznek ahhoz, hogy rávegyék a felhasználókat arra, hogy elárulják a jelszavukat, vagy hogy a hálózat rendszergazdájától jelszavakat tudjanak meg. Egyszerűen felhívják a segélyvonalat, egy kicsit elveszettnek tettetik magukat, és azt mondják, hogy elfelejtették a jelszavukat. Ez ostobán hangozhat, de a támadók sokszor rengeteg erőfeszítést megtakaríthatnak vele, és általában ez az első dolog, amivel próbálkoznak. Minden vállalatnak és intézménynek világosan utasítania kellene a számítógépes szakembereit, hogy senkinek ne adjanak ki jelszavakat anélkül, hogy meggyőződnének a kérés jogosságáról.

Ahogy az óra későbbi részében majd megtanuljuk, a támadó végső célja az, hogy rendszergazdai szintű jogosultságokhoz jusson. Minden jelszót védeni kell, mert bármilyen hozzáférés eredményezhet rendszergazdai hozzáférést, a rendszergazdai fiókokat azonban különösképpen védeni kell a feltöréstől. A rendszergazda felhasználóneve is olyan védelmi vonal, amely útját állhatja a támadásnak. A legtöbb számítógéprendszer jól dokumentált és jól ismert alapértelmezett rendszergazdai fiókkal rendelkezik. Ha a támadó ismeri az operációs rendszert, máris jó úton jár a rendszergazdai jogosultságok megszerzése felé, csak tudnia kell a rendszergazdai fiók felhasználónevét. A szakértők ezért azt javasolják, hogy változtassuk meg a rendszergazdai fiók felhasználónevét.

Trójai programok

A számítógépes támadók egyik gyakran használt eszközét jelentik az úgynevezett trójai programok. A trójai program általánosságban egy olyan számítógépes program, amely látszólag ártalmatlan műveleteket végez, miközben a színfalak mögött láthatatlan és rosszindulatú tevékenységet folytat. A trójai programok egyik korai változata volt a hamis bejelentkező képernyő. Az ilyen ablakok pontosan úgy néztek ki, mint

a rendszer által használt bejelentkező képernyők, de amikor a felhasználó megpróbált bejelentkezni, a támadó elfogta a felhasználónevét és a jelszavát, és egy a támadó számára elérhető titkos helyre mentette (lásd a 22.1. ábrát).



22.1. ábra

Jelszavak ellopása trójai bejelentkező program segítségével

Ahogy kitalálhatjuk, a jelszavak ellopásának ezt a módszerét olyan nyilvános környezetekhez tervezték, mint egy számítógéplabor, ahol a felhasználók közösen használják a terminálokat vagy munkaállomásokat. Az utóbbi években azonban az operációs rendszerek felkészültebbé váltak arra, hogy megakadályozzák, de legalábbis észleljék a jelszóelfogásnak ezt a módját.

Nem minden trójai program jelszavak elfogását szolgálja, és nem minden trójai olyan otromba, mint a fentebb leírt példában szereplő. Az Interneten a trójai programok számos más típusa is hozzáférhető. Egyesek játéknak vagy rendszer-segédprogramoknak álcázzák magukat; sokat ingyenes vagy kipróbálható szoftverként terjesztenek az Interneten. Az ilyen támadások ellen a legjobb védekezés az, ha körültekintően töltünk le programokat. Mielőtt letöltenénk és telepítenénk egy ingyenes segédprogramot, olvassuk el a program leírását, és nézzünk utána az Interneten, hogy nincsenek-e vele kapcsolatos biztonsági figyelmeztetések. Gondoljunk Kasszandra hercegnőre, aki Kr.e. 800-ban így jósolta meg egy különösen veszélyes trójai faló érkezését városa kapujához: „Óvakodj a görögök ajándékától".

Találgatás

Egyes jelszavak annyira egyszerűek vagy rosszul formáltak, hogy a támadó könnyen kitalálhatja őket. Meglepődnénk, ha tudnánk, hány felhasználónak egyezik meg a jelszava a felhasználónevével. Vannak olyanok is, akik egy utca nevét, a leánykori nevüket, vagy az egyik gyerekük nevét használják jelszóként, mások pedig olyan könnyen kitalálható karakterkombinációkat alkalmaznak, mint az `123456`, az `abcde` vagy a `zzzzzz`.
Még ha a támadó alig tud is valamit a felhasználóról, a rossz jelszavakat akkor is sokszor kitalálhatja. Valójában a támadónak már nem is kell találgatnia, mert léteznek olyan eszközök, amelyek automatizálják a jelszókitalálás folyamatát. Az ilyen eszközök a gyakran használt karakterkombinációk listáján haladnak végig, sőt egyesek még egy szótárat is felhasználnak, hogy egy adott nyelv összes lehetséges szavát vagy nevét kipróbálják. Akár több ezer próbálkozásra is szükség lehet, de a számítógépek ezt gyorsan elvégzik.

Jelszóelfogás

A csomagszimatolók és más eszközök, amelyek a hálózati forgalmat figyelik, könnyen képesek elfogni a hálózaton sima szövegként (tehát titkosítatlan formában) átvitt jelszavakat. Sok klasszikus TCP/IP-segédprogramot – például a Telnetet, az r* programokat vagy az SNMP-t (lásd a 15. fejezetben) – úgy terveztek, hogy a jelszavakat tisztán szöveges formában adja át. Ezeknek a programoknak az újabb változatai időnként már lehetővé teszik a jelszavak titkosítását, illetve a biztonságos csatornán keresztüli működést (lásd a 23. fejezetet). Eredeti formájukban azonban a sima szöveges jelszavak ezeket az alkalmazásokat reménytelenül alkalmatlanná teszik egy olyan nyílt és ellenséges környezetben való használatra, mint az Internet.

A sima szöveges jelszavak még egy zárt környezetben (például egy céges hálózaton) sem igazán biztonságosak. Egyes szakértők becslése szerint százból egy vállalati alkalmazott mindig akad, aki aktívan fenyegeti a hálózat biztonságát. Egy százalék nem sok, de egy 1000 felhasználóval rendelkező hálózatban ez már 10 olyan felhasználót jelent, aki rá szeretné tenni a kezét valaki másnak a jelszavára.

A jelszavak titkosítására számos módszer létezik. A titkosított jelszavak sokkal jobbak a sima szöveges jelszavaknál, de a titkosításnak is megvannak a maga korlátai. Az olyan eszközök, mint az LC5 vagy a John the Ripper, szótár vagy nyers erő (próbálgatás) alkalmazásával képesek visszafejteni a titkosított jelszavakat.

Az Interneten tevékenykedő támadók el tudják fogni a titkosított jelszavakat tartalmazó csomagokat, és az említett jelszófejtő segédprogramokkal fel tudják fedni a jelszavakat. Az újabban kifejlesztett titkosított csatornák, például az SSL vagy az IPsec (lásd a 23. órát), viszont jelentősen megemelik a lécet a TCP/IP-vonalon érzékeny információk, például jelszavak megszerzése céljából hallgatózni kívánó támadók számára.

Ha a támadó valamilyen hozzáférést szerzett a rendszerhez, többféle lehetőség is a rendelkezésére áll arra, hogy más rendszerjelszavakat is elfogjon vagy felfedjen, beleértve a rendszergazdai jelszavakat. Egyes eszközök lehetővé teszik a támadónak, hogy rögzítse és naplózza a felhasználók billentyűleütéseit, amikor jelszavakat billentyűznek be. Ezen kívül a támadó jelszó segítségével hozzáférhet titkosított rendszerfájlokhoz is, és kapcsolat nélkül, a szokásos jelszófejtő eljárásokat alkalmazva elemezheti azokat, hogy további jelszavakhoz jusson hozzá.

Mi tehetünk az azonosítók elleni támadások megakadályozása érdekében?

Az azonosítók megszerzésére irányuló támadások elleni legjobb védelem a folyamatos éberség. A hálózatok különféle stratégiákat alkalmaznak, hogy csökkentsék az esélyét a jelszavak feltörésének. Lássunk néhányat a nyilvánvalóbb irányelvek közül:

1. Határozzunk meg erős és világos jelszóházirendet a felhasználóink számára. Figyelmeztessük őket, hogy milyen veszélyekkel jár, ha kiadják a jelszavukat más felhasználóknak, leírják azt egy ragadós cetlire, amit aztán valahová az asztaluk mellé ragasztanak, vagy ha a jelszavukat egy fájlban tárolják.
2. Állítsunk be minden számítógépet úgy, hogy kötelezővé tegye a jelszóházirend betartását. Rendszeres időnként változtatassuk meg a felhasználókkal a jelszavukat, és határozzuk meg azok minimális hosszúságát (általában 6-8 karaktert szoktak megadni). Ne engedjük, hogy a felhasználók olyan jelszavakat használjanak, mint a kutyájuk vagy a gyerekük neve. Tulajdonképpen egyáltalán nem szabad, hogy a jelszavak értelmes szavak, kifejezések vagy nevek legyenek. Minden jelszóban legyenek betűk és számok, és legalább egy nem alfanumerikus karakter, ami nem az első vagy az utolsó karakter. A jelszókitalálásos támadások megelőzése érdekében állítsuk be úgy a számítógépeket, hogy egy bizonyos számú sikertelen belépési kísérlet után letiltsák a fiókokat.
3. Gondoskodjunk róla, hogy a jelszavak soha ne utazzanak sima szövegként nyilvános vonalakon keresztül. Ha lehetséges, a belső hálózatunkon se vigyünk át sima szöveges jelszavakat, különösen ha nagy hálózatról van szó.

Egyes rendszerek különféle eljárásokkal szabályozzák a jelszavak számát, amelyekre az egyes felhasználóknak emlékezniük kell. A Microsoft-hálózatokban egy jelszógyorsítótárat találunk, valamint egységes hálózati bejelentkezést a tartomány biztonsági rendszerén keresztül. A Unix rendszerek olyan megoldásokat kínálnak, mint a Kerberos-hitelesítés (lásd a 23. fejezetet). Ezek az eljárások egyes környezetekben hasznosak a jelszavak elburjánzásának megakadályozásában, az egységesített bejelentkezési eljárások hátránya ugyanakkor, hogy ha a támadónak egyetlen jelszót is sikerül megszereznie, akkor az adott felhasználó erőforrásaihoz korlátlanul hozzáférhet.

A jelszavak titkosítással történő védelméről a 23. órán bővebben is beszélünk.

Hálózatszintű támadások

Ahogy a 6. órán megtanultuk, a hálózati alkalmazások elérésének kezelése a TCP/IP-verem szállítási rétegében a kapunak (port) nevezett logikai csatornákon keresztül történik. A támadók gyakran úgy szereznek hozzáférést egy rendszerhez, hogy keresnek egy nyitott kaput, amely egy hálózati kapcsolatokra váró hálózati szolgáltatáshoz vezet. Egyes esetekben a szolgáltatás alapértelmezés szerint is futhat, anélkül, hogy rendszer tulajdonosa egyáltalán tudna róla, míg máskor a szolgáltatás beállítása lehet hibás, esetleg lehetővé teheti a hozzáférést egy alapértelmezett vagy névtelen felhasználói fiókon keresztül. Az olyan pásztázóeszközök, mint az Nmap vagy a Nessus, automatizálják a nyitott kapuk keresésének folyamatát. Ezeket a pásztázókat a támadók (akik réseket keresnek, hogy hozzáféréshez jussanak) és az informatikus szakemberek (akik azért keresik a réseket, hogy betömjék azokat, és megakadályozzák a hozzáférést) egyaránt használják. Más, speciálisabb eszközök adott hálózati protokollok és szolgáltatások rései után kutatnak. Sok esetben egy nyitott kapu puszta léte nem elég ahhoz, hogy a támadó bejusson, de lehetőséget teremt arra, hogy a támadó alkalmazásszintű támadást indítson, a kapun figyelő szolgáltatás ismert sebezhető pontjait kiaknázva.

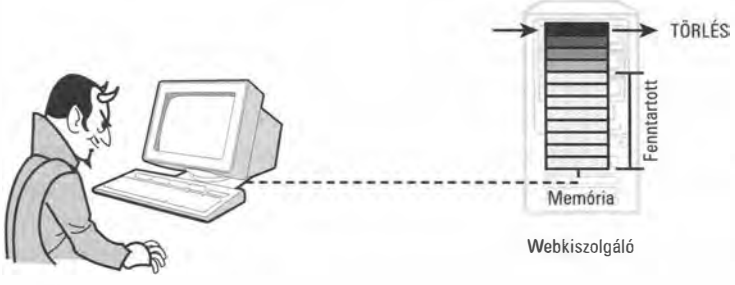
A pásztázóprogramok szó szerint folyamatosan futnak az Interneten, az IP-címek teljes tartományát bejárva, nyitott kapuk és védtelen szolgáltatások után kutatva. Ahogy a 10. órán megtanultuk, a tűzfalak egyik fontos feladata, hogy lezárják a hozzáférést, hogy a hálózati pásztázóprogramok ne tudjanak információt szerezni a hálózaton működő szolgáltatásokról.

Más hálózatszintű támadási módszerek is léteznek, amelyeket a nyílt Interneten alkalmaznak a TCP/IP-forgalom elfogására és feltörésére. A munkamenet-eltérítés például egy olyan haladó eljárás, amely a TCP protokoll egyik hiányosságát aknázza ki. Ahogy a 6. órán megtanultuk, a TCP protokoll egy munkamenetet nyit a hálózati állomások között. A munkamenet-eltérítés során a támadó lehallgat egy TCP-munkamenetet, és olyan csomagokat szúr be az adatfolyamba, amelyek látszólag a TCP-munkamenet részei. A támadó ezzel a módszerrel parancsokat juttathat be az eredeti munkamenet biztonsági környezetébe. A munkamenet-eltérítést gyakran alkalmazzák arra, hogy a rendszert jelszavak felfedésére vagy megváltoztatására vegyék rá.

Természetesen a támadó nem saját kezűleg, menet közben állítja össze a hamis TCP-szegmenseket – a munkamenet-eltérítés különleges eszközöket kíván. Az egyik hírhedt eszköz, amelyet munkamenet-eltérítésre használnak, a Juggernaut nevű ingyenes alkalmazás. A Juggernaut egy helyi hálózaton hallgatózik, és adatbázist készít a TCP-kapcsolatokról. A támadó figyelheti a TCP-forgalmat, hogy visszajátssza a kapcsolati előzményeket, vagy eltérítsen egy aktív munkamenetet tetszőleges parancsok befecskendezésével. Ahogy majd a 23. órán látni fogjuk, a munkamenet-eltérítéssel és más protokoll alapú támadásokkal szemben a legjobb védekezés az, ha virtuális magánhálózat (VPN) vagy valamilyen más titkosított kommunikációs forma segítségével biztosítjuk a munkamenetet.

Alkalmazásszintű támadások

Arra számítanánk, hogy ha a szoftvert megfelelően állítjuk be, és a jelszavakat távol tartjuk az ellenségtől, nem lehet semmilyen gondunk az internetes támadókkal. Sajnos azonban a valóság ennél kissé bonyolultabb. Az Interneten ma futó programok közül sokat évekkel ezelőtt írtak – akkor, amikor a behatolás művészete még gyerekcipőben járt –, ezért gyakran olyan kódrészleteket tartalmaznak, amelyek eredendően nem biztonságosak. Még a ma készített programokat is túl gyakran sietve írják olyan programozók, akiknek a tudása és a tapasztalata erősen különböző, a támadók pedig különféle megoldásokat dolgoztak ki a rendszerek feltörésére a nem biztosított programkódok kiaknázásával.



22.2. ábra

A tártúlcsordulásos támadás túlteliti a programbemenet számára fenntartott memóriaterületet, ami a program összeomlását, furcsa viselkedését vagy a támadó parancsainak végrehajtását okozza

Az alkalmazásszintű támadási módszerek egyik legnépszerűbbike az *átmeneti tárak túlcsordulásának* (buffer overflow) előidézése. Amikor egy számítógép adatokat kap egy hálózati kapcsolaton keresztül (sőt akár a billentyűzetről), elegendő memóriaterületet kell lefoglalnia ahhoz, hogy a teljes adathalmazt fogadhassa. Ezt a fogadóterületet hívják *átmeneti tárnak* (buffer). Ha a felhasználótól kapott bemenet meghaladja az átmeneti tár méretét, furcsa dolgok történhetnek. Ha a bemenet kezelése nem megfelelően történik, a táron túlcsorduló adatok a processzor végrehajtási területére kerülhetnek, ami azt jelenti, hogy a tártúlcsorduláson keresztül a számítógépnek küldött parancsokat ténylegesen végre lehet hajtani (lásd a 22.2. ábrát), és ezek a parancsok az adatokat fogadó alkalmazás jogosultságaival hajtódnak végre. Más tártúlcsordulásos támadások azt a tényt használják ki, hogy egyes alkalmazások olyan, magasabb szintű biztonsági környezetben futnak, ami akkor is aktív marad, ha az alkalmazás váratlanul leáll.

A tártúlcsordulás elkerülése érdekében az alkalmazásoknak valahogyan biztosítaniuk kell az adatok fogadását és azok méretének ellenőrzését, mielőtt az adatokat beszúrnák egy alkalmazás átmeneti tárába. A megoldás nagyrészt a helyes programozás függvénye. A rosszul megtervezett alkalmazások különösen könnyen áldozatul eshetnek a tártúlcsordulásos támadásoknak.

Egyes híres és népszerű hálózati alkalmazások sebezhetők a tártúlcsordulásos támadásokkal szemben. Ezeket a hiányosságokat jól ismerik szerte az Interneten, ezért a támadók pontosan tudják, hol és hogyan kell támadást indítaniuk. A Unix alapú Sendmail levélkiszolgáló gyakori célpontja a tártúlcsordulásos támadásoknak, de az utóbbi években a Microsoft Internet Information Server (IIS) programja és más termékei is sokszor áldozatul esnek az ilyen támadásoknak. Amikor egy gyártó felfedez egy tártúlcsordulásra hajlamos pontot egy programban, általában javítócsomagot („foltot") bocsát ki hozzá, amely kijavítja a problémát. Mivel a tártúlcsordulással szemben sebezhető pontok nagy nyilvánosságot kapnak, a gyártók igyekeznek gyorsan kijavítani a szoftvert, amint felfedeznek egy hiányosságot. Nem ritka, hogy egy gyártó a biztonsági probléma felfedezése után napokon vagy akár órákon belül közzétesz egy javítócsomagot. A jó rendszergazdák ezenkívül éberen figyelik az olyan szervezetektől érkező biztonsági riasztásokat, mint a Common Vulnerabilities and Exposures projekt (`http://cve.mitre.org`), hogy tudják, mikor és hol szerezhetik be a rendszerükhöz szükséges legfrissebb javítócsomagokat. Az olyan szervezetek, mint a SANS (`http://www.sans.org`), elektronikus hírleveleket is közzétesznek, amelyek a legújabb biztonsági fenyegetésekről tájékoztatnak.

A tártúlcsorduláshoz hasonló problémákra részben a helyes programozás kínál megoldást – nem csak a nagy gyártók programjaiban, hanem az otthon, webfejlesztők és informatikus munkatársak által készített parancsfájlokban is. A megoldás másik részét a rendszer naprakészen tartása jelenti az összes javítócsomag és frissítés telepítésével. Egyes operációs rendszerek lehetővé teszik a tártúlcsordulás előidézésével próbálkozó távoli felhasználók jogosultsági körének korlátozását. Ha lehet, ne hagyjuk a hálózati alkalmazásokat `root` vagy rendszergazdai jogosultságokkal futni. (Előfordulhat persze, hogy nincs választási lehetőségünk.) Azoknak a programoknak az esetében, amelyeknek a működéséhez magas jogosultsági szint szükséges, a Unix/Linux `chroot`-jához hasonló alkalmazásokkal hozhatunk létre korlátozott biztonsági környezetet, amely megakadályozza, hogy a támadók hozzáférjenek a rendszer többi részéhez.

Gyökérszintű hozzáférés

A hálózati támadók Szent Grálja mindig a rendszergazdai vagy *gyökérszintű* (root) hozzáférés a rendszerhez. A `root` hozzáféréssel rendelkező felhasználó bármilyen parancsot végrehajthat, illetve bármilyen fájlt megtekinthet. Ha `root` hozzáférésünk van, lényegében azt tehetünk a rendszerrel, amit csak akarunk. Maga a „root" kifejezés a Unix világából ered, de a rendszer vezérlésére jogosult kiemelt fiók elgondolását minden gyártónál és platformon megtaláljuk. Windows-hálózatokon ezt a fiókot hívják rendszergazdai fióknak (Administrator fiók, felügyeleti fiók).

Miután a támadó belülre került, az egyik első dolog, amit tenni szokott, hogy feltölt egy úgynevezett *rootkit*-et („gyökérkészletet"). A rootkit olyan eszközökből áll, amelyekkel maradandóbb állás építhető ki a rendszeren. Az eszközök egy része új rendszerek és új fiókok feltörésére szolgál, míg másokat arra terveztek, hogy elrejtsék a támadó jelenlétét

a rendszeren. Ezek az elfedőeszközök sokszor olyan, szabványos hálózati segédprogramok megbütykölt változatai, mint a `netstat`, vagy olyan alkalmazások, amelyek eltüntetik a támadó nyomát a rendszer naplófájljaiból. A rootkit más eszközei a hálózat feltérképezésében vagy további jelszavak elfogásában segítenek a támadónak. Egyes gyökérkészletek még arra is módot adnak a támadónak, hogy magát az operációs rendszert módosítsa.

A támadó ezt követően egy vagy több hátsó ajtót próbál létesíteni a rendszeren, vagyis olyan titkos bejáratokat a rendszerbe, amelyeket a hálózat rendszergazdája nehezen fedezhet fel. A hátsó ajtó célja, hogy lehetővé tegye a támadónak, hogy kikerülje a szokásos interaktív hozzáférést körülvevő naplózó és figyelő eljárásokat. A hátsó ajtó lehet egy rejtett fiók, de lehetnek egy eredetileg csak korlátozott hozzáféréssel rendelkező fiókhoz társított rejtett jogosultságok is. Egyes esetekben a hátsó ajtóhoz vezető út olyan szolgáltatásokat is tartalmazhat, mint a Telnet, olyan szokatlan kapuszámokhoz rendelve, amelyekre a helyi rendszergazda nem számít.

Miután a támadó feltöltötte a szükséges eszközöket, és megtette a nyomok eltüntetéséhez, valamint a későbbi visszatéréshez szükséges intézkedéseket, a következő lépés azoknak a kártevő műveleteknek a végrehajtása (fájlok vagy hitelkártya-adatok ellopása, a rendszer levélszemét-küldőként történő beállítása), amelyeket a támadó kieszelt a hálózat számára. A támadók másik célja azonban az, hogy felkészüljenek a következő támadásra. Egy körültekintő támadó soha nem hagy olyan nyomot maga után, ami elvezethet hozzá, ezért általában egy már feltört rendszerről indít támadást. Egyes támadók több távoli rendszer láncolatán keresztül tevékenykednek, mert ez a megoldás szinte lehetetlenné teszi a támadó valódi helyének megállapítását.

Adathalászat

A tűzfalak és titkosítási eljárások használata, illetve az egyéb biztonsági intézkedések megnehezítik a támadóknak, hogy egyszerűen hívatlanul besétáljanak egy hálózatba, de válaszul a biztonsági rendszerek megkerülésére eljárások új nemzedékét dolgozták ki. Az egyik ilyen fontos új stratégia a gyanútlan felhasználók bevonása a támadásba egy csalinak használt hamis hivatkozáson, elektronikus levélen vagy weboldalon keresztül. Ezt a fajta támadást hívják *adathalászatnak* (phishing). Az adathalászatra példa, amikor egy levél arra kéri a felhasználót, hogy jelentkezzen be egy online bankba, és frissítse a fiókja adatait, miközben a megadott cím egy hamis webhelyre vezet, amely a támadó irányítása alatt áll.

Az adathalász támadások gyakran azt a tényt használják ki, hogy egy hivatkozás szövege független a tényleges URL-től, amelyre a hivatkozás mutat. Ahogy a 17. fejezetben megtanultuk, a webfejlesztő így határozhat meg egy hiperhivatkozást:

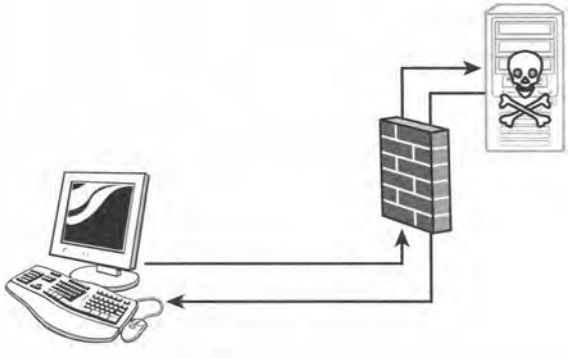
```
<a href="http://www.MyBank.com/">MyBank</a>
```

Ebben az esetben a „MyBank" szó egy olyan hivatkozásként jelenik meg, amely a `http://www.MyBank.com/` címre vezet. De mi történik, ha az erkölcsileg kihívásokkal küszködő programozó így kódolja a hivatkozást:

```
<a href="http://www.NOT_MyBank_$$&%%%??!!!.biz/">MyBank</a>
```

A hivatkozás ebben az esetben is a MyBank szöveggel jelenik meg, de egy egészen más webhelyre mutat. Ha körültekintőek vagyunk, a webböngészőnk címsávjában vagy az egeret a hivatkozás fölé mozgatva egy szövegbuborékban néha megláthatjuk, hogy adathalász URL-ről van-e szó. A legjobb megoldás, ha soha nem kattintunk kéretlen levelekben található hivatkozásokra, és soha nem adunk meg az Interneten pénzügyi információkat, hacsak nem magunk kezdeményeztük a műveletet, és biztosak nem vagyunk benne, hogy ott vagyunk, ahol akartunk.

Más, összetettebb adathalászati módszerek nehezebben észlelhetők és felderíthetők. A külső parancsfájl befecskendezése (cross-site scripting) nevű eljárás a böngésző biztonsági rendszerét egy olyan kód befecskendezésével kerüli meg, amely egy olyan rosszindulatú parancsfájlt indít el, amelyet nem lehet egyszerűen visszakövetni addig az oldalig, amelyet a felhasználó lát.



22.3. ábra

Egy otthoni tűzfal, amely letiltja a kívülről érkező kapcsolódási kísérleteket, gyakran hatástalan, ha a felhasználó kezdeményezi a kapcsolat felvételét egy hamis webkiszolgálóval

A támadások gyanútlan felhasználókon keresztül történő elindítása nem csupán egy egyszerű trükk arra, hogy hamis webhelyekre mutató hivatkozásokkal irányítsák a böngészőt. A tűzfalakhoz hasonló eszközöket ugyanis elsősorban a kívülről érkező támadások kivédésére tervezték. Azzal, hogy egy ártatlan felhasználó kezdeményezi a kapcsolatot, a támadó számos óvintézkedést megkerülhet, amelyet a hálózat biztonsági rendszere alkalmaz (lásd a 22.3. ábrát). A böngésző és a tűzfal nem tudja egyszerűen kideríteni, hogy a kapcsolat különbözik-e bármely más, külső webhellyel felvett kapcsolattól. Ha pedig a kapcsolat létrejön, a támadó különféle módszerekkel feltörheti a biztonsági rendszert, amire nem lenne lehetősége, ha a támadást a tűzfalon kívülről

indítaná. Ezt a fajta támadást még a hálózati címfordítás (Network Address Translation, NAT, lásd a 12. fejezetben) sem képes kivédeni, hiába rendel a felhasználó rendszeréhez egy elvileg el nem téríthető IP-címet, mert a tűzfal egyszerűen lefordítja a munkamenet forgalmát, ahogy bármely más HTTP-kapcsolat esetében tenné.

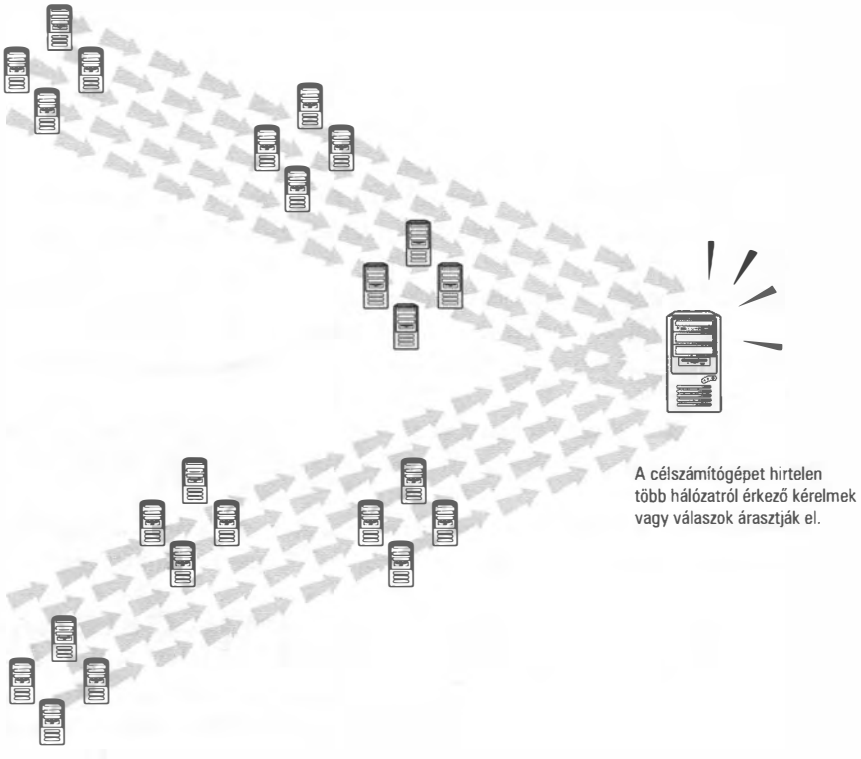
Az ilyen, felhasználó által kezdeményezett támadások lehetősége az egyik oka, amiért a biztonsági szakemberek nem bíznak különösebben a boltban megvásárolható, otthonra szánt tűzfaleszközökben. A szakértők inkább azokat a kifinomultabb tűzfaleszközöket részesítik előnyben, amelyek rugalmasabb szabályozást és szűrést tesznek lehetővé.

Elárasztásos támadások

Az Interneten ma az egyik „legmenőbb" támadási forma az elárasztásos vagy szolgáltatás-megtagadást (denial-of-service, DoS) előidéző támadás. Ha megkezdődtek, a DoS-támadásokat szinte lehetetlen megállítani, mert a támadótól nem igényelnek semmilyen különleges jogosultságot a rendszeren. A DoS-támadások célja a rendszer megbénítása olyan sok kérelemmel, ami felemészti a rendszer erőforrásait, és a minimumra csökkenti a teljesítményét. Jelentős méretű elárasztásos támadást hajtottak már végre az Egyesült Államok kormányzati webhelyei, illetve a főbb internetes keresők ellen is.

A legveszélyesebb DoS-támadás az úgynevezett elosztott elárasztás. Az elosztott elárasztás során a támadó több távoli számítógép segítségével utasít más távoli számítógépeket összehangolt támadásra. Egy egyetlen IP-cím elleni támadásban néha számítógépek százai vagy ezrei vehetnek így részt.

Az elárasztásos támadások gyakran a TCP/IP szabványos kapcsolati segédprogramjait használják. A hírhedt Smurf támadás például a `ping` segédprogramra támaszkodik (lásd a 14. fejezetet), és visszhangkérések áradatát zúdítja az áldozatra (22.4. ábra). A támadó irányított többes címzésen keresztül a teljes hálózatnak visszhangkéréseket küld, és a kérelmek forráscímét úgy álcázza, hogy úgy tűnjön, hogy azok az áldozat IP-címéről érkeznek. Ezt követően a hálózaton található valamennyi számítógép egyszerre válaszol a visszhangkérésre. A Smurf támadás eredménye az, hogy a támadó eredeti visszhangkérése sok-sok visszhangkéréssé fokozódik a hálózaton. Ha a támadó egyidejűleg több hálózaton is elindítja a folyamatot, visszhangválaszok hatalmas árhullámát keltheti, amely megbénítja az áldozat rendszerét.



22.4. ábra
Elárasztásos támadás

Összefoglalás

Ebben az órában az internetes támadók különféle módszereit írtuk le, amelyekkel hozzáférést szerezhetnek egy hálózathoz. Tanultunk az azonosítók elleni, a hálózatszintű, illetve az olyan alkalmazásszintű támadásokról, mint az átmeneti tárak túlcsordulásának előidézése, és megismertük a gyökérkészleteket, a hátsó ajtókat, az adathalászatot, valamint az elárasztásos támadásokat.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Hátsó ajtó** – Rejtett bejárat egy számítógéprendszerbe.
- **Tártúlcsordulás** – Ezt idézik elő azok a támadások, amelyek során a támadó túltelíti egy alkalmazás átmeneti tárát, hogy rosszindulatú parancsokat adhasson ki a rendszernek.
- **Elárasztásos támadás (szolgáltatás-megtagadásos támadás, DoS)** – A rendszer-erőforrások felemésztésével az áldozat rendszerének megbénítását célzó támadási forma.
- **Levélféreg** – Rosszindulatú parancsfájl vagy program, amelyet elektronikus levélben terjesztenek.
- **Adathalászat** – Hamis hivatkozás, üzenet vagy weboldal segítségével egy ártatlan felhasználó rávétele arra, hogy kapcsolatot létesítsen egy rosszindulatú webhellyel.
- **Gyökérszintű hozzáférés** – A hozzáférés legmagasabb szintje egy számítógéprendszeren. A gyökérszintű vagy rendszergazdai hozzáférés szinte korlátlan vezérlést tesz lehetővé.
- **Rootkit (gyökérkészlet)** – Támadók által használt eszközkészlet, amellyel kibővíthetik és álcázhatják a rendszer feletti uralmukat.
- **Szkriptkölyök** – Fiatal, általában tinédzserkorú internetes behatoló, aki leginkább előregyártott, az Interneten elérhető parancsfájlokkal és eszközökkel dolgozik.
- **Munkamenet-eltérítés** – Támadási módszer, amely lehetővé teszi a támadónak, hogy rosszindulatú csomagokat szúrjon be egy meglevő TCP-munkamenetbe.
- **Trójai program** – Olyan program, amely látszólag ártalmatlan műveleteket végez, miközben a színfalak mögött láthatatlan és rosszindulatú tevékenységet folytat.

23. ÓRA

A TCP/IP biztonsága

A fejezet tartalmából:

- Titkosítás
- Tanúsítványok
- A TCP/IP biztosítása

Ahogy az előző órán megtanultuk, a jogosulatlan felhasználók már arra is hatalmas erőfeszítéseket tesznek, hogy elfogják a kommunikációs forgalmat, és belopózzanak mások hálózatára. A szakemberek egyre jobb módszereket dolgoznak ki a TCP/IP-kommunikáció elrejtésére, hogy a támadók ne leshessenek ki titkokat a hálózaton. Ebben az órában a TCP/IP biztonságossá tételének fontosabb módszerei közül mutatunk be néhányat. Az óra végeztével a következőkre leszünk képesek:

- Meg tudjuk határozni a *titkosító algoritmus és a titkosító kulcs* kifejezések jelentését.
- El tudjuk megyarázni a szimmetrikus és aszimmetrikus titkosítás közötti különbségeket.
- Le tudjuk írni a digitális aláírás és a digitális tanúsítványok szerepét.
- Le tudjuk írni a TLS/SSL és az IPSec TCP/IP-biztonsági protokollrendszerek működését.
- El tudjuk magyarázni, mi az a virtuális magánhálózat, és hogyan működik.
- Le tudjuk írni a Kerberos hitelesítési eljárás menetét.

Titkosítás

Egy nyilvános hálózaton áthaladó védtelen adatcsomagot könnyen el lehet fogni és el lehet olvasni. Egyes esetekben az adatok felhasználókkal vagy jelszavakkal kapcsolatos információkat tartalmazhatnak, máskor pedig olyan más érzékeny információkat, amelyeket nem szeretnénk, ha bárki más látna – például hitelkártyaszámokat vagy vállalati titkokat. Még ha az adatok nem is különösebben titkosak, sok felhasználó érthetően kényelmetlenül érzi magát, ha attól kell félnie, hogy valaki lehallgatja az általa folytatott elektronikus kommunikációt.

Az óra későbbi részében terítékre kerülő biztonsági eljárások arra szolgálnak, hogy a hálózatot biztonságosabbá tegyék. Ezek közül az eljárások közül sok titkosítást alkalmaz. A *titkosítás* az adatok szisztematikus módosításának folyamata, amelynek célja az adatok olvashatatlanná tétele a jogosulatlan felhasználók számára. Az adatokat a küldő *titkosítja*, majd azok kódolt, elolvashatatlan formában haladnak át a hálózaton, hogy aztán a fogadó számítógép *visszafejtse* őket, hogy el tudja olvasni.

A titkosítás valójában egyáltalán nem igényel számítógépet. Titkosítási módszerek évszázadok óta léteznek. Amióta az ember titkos üzeneteket ír, mindig keresi azokat a kódolási formákat vagy trükköket, amelyekkel megőrizheti az üzenetek titkosságát. A számítógépek korában azonban a titkosítás sokkal kifinomultabbá vált, mert a számítógépek hatalmas, bonyolult számokkal is megbirkóznak. A legtöbb számítógépes titkosító algoritmus nagy prímszámokkal dolgozik. Maguk az algoritmusok bonyolult matematikai szabályokon alapulnak, és nem túlzás azt állítani, hogy a titkosító algoritmusokat készítő szakemberek többsége diplomás matematikus vagy informatikus.

A titkosítás alapvetően fontos részét képezi szinte minden TCP/IP-biztonsági eljárásnak. A következőkben a titkosítással kapcsolatos legfontosabb fogalmakat tekintjük át. Ahogy haladunk előre az órán, fontos, hogy észben tartsuk, hogy a biztonsági rendszernek valójában több célja is van, és a biztonsági eljárások többféle igényt elégítenek ki. Ennek a résznek az elején a bizalmasságról (az adatok titkosságának megőrzéséről) beszéltünk, de egy biztonsági rendszernek a következőkre is gondolnia kell:

- **Hitelesítés** – Meg kell győződnie arról, hogy az adatok valóban abból a forrásból erednek, ahonnan állítják.
- **Adatépség** – Meg kell bizonyosodnia róla, hogy az adatokat átvitel közben nem piszkálták meg.

A titkosítási eljárások a hitelesítésről, az adatok épségéről és a bizalmasságról egyaránt gondoskodnak.

Ebben az órában azt vesszük górcső alá, hogy miként védhetjük meg a TCP/IP protokollokat a hallgatózástól, az elfogástól és a módosítástól. Egy hálózat általános biztonságának szempontjából ugyanakkor más tényezők is fontosak. A TCP/IP-hálózatok biztonságáról további ismereteket a 10. és 22. fejezetben találhatunk.

Algoritmusok és kulcsok

Ahogy az előző részben megtanultuk, a titkosítás az az eljárás, amelynek során az adatokat olvashatatlanná teszik minden és mindenki számára, ami vagy aki nem ismeri a titkosító kódot nyitó varázsigét. Ahhoz, hogy a titkosítás működhessen, a kommunikációban részt vevő két félnek az alábbiakkal kell rendelkeznie:

- egy eljárással, amellyel az adatok olvashatatlanná tehetők (titkosítás), és
- egy eljárással, amely az olvashatatlan adatokat képes visszaalakítani az eredeti, olvasható formájukra (visszafejtés).

Amikor a programozók először fogtak titkosító programok írásába, rájöttek, hogy a következő problémákat kell megoldaniuk:

- Ha minden számítógép pontosan ugyanazt az eljárást alkalmazza az adatok titkosítására és visszafejtésére, a program nem lesz kellően biztonságos, mert a hallgatózóknak csak szerezniük kell egy példányt a programból, és máris visszafejthetik az üzeneteket.
- Ha minden számítógép teljesen eltérő és egymással kapcsolatban nem álló eljárásokat használ az adatok titkosítására és visszafejtésére, akkor minden számítógépen teljesen eltérő és egymással kapcsolatban nem álló programra van szükség, a kommunikálni kívánó pároknak pedig külön szoftverre. Ez rendkívül költséges, és a nagy, különböző rendszerekből álló hálózatokon kezelhetetlen lenne.

Ezek a problémák megoldhatatlannak tűnnek, de a titkosítási eljárásokat feltaláló nagy koponyák mégis gyorsan találtak rájuk megoldást – mégpedig azt, hogy az adatok titkosítására és visszafejtésére használt eljárást egy szabványos, megismételhető részre (ami mindig azonos) és egy egyedi részre (ez kényszeríti ki a titkos kapcsolatot a kommunikáló felek között) bontják. A titkosítási eljárás szabványos részét hívjuk titkosító algoritmusnak. A *titkosító algoritmus* lényegében matematikai lépések sorozata, amelyekkel az adatokat olvashatatlan formára alakítjuk. Az eljárás egyedi, titkos része a *titkosító kulcs*. A titkosítás tudománya rendkívül bonyolult, de jelen tárgyalás céljainak megfelel, ha a kulcsot úgy képzeljük el, mint egy nagy számot, amelyet változóként használunk az algoritmusban. A titkosítási eljárás eredménye a kulcs értékétől függ, tehát ha a kulcs értékét titokban tartjuk, a jogosulatlan felhasználók akkor sem lesznek képesek elolvasni az adatokat, ha rendelkeznek a szükséges visszafejtő szoftverrel.

A jó titkosító algoritmusok egyediségét és homályosságát nem lehet túlbecsülni, ennek ellenére a következő példa jól szemlélteti a kulcs és az algoritmus működési elvét. Egy férfi nem szeretné, ha az anyja tudná, hogy mennyit fizet a bútorokért. Tisztában van vele, hogy az anyja érdeklődik a matematika iránt, ezért nem akarja megkockáztatni, hogy egy egyszerű tényezőt vagy szorzót használjon a valódi értékek elfedésére, nehogy az anyja rájöjjön a képletre. A barátnőjével megbeszéli, hogy ha az anyja látogatóba jön, és rákérdez egy bútor árára, a valódi árat egy hasraütésszerűen kitalált számmal elosztja, az eredményt megszorozza kettővel, majd hozzáad 10 dollárt. Más szavakkal, a következő algoritmust szándékozik alkalmazni:

$$\frac{\text{(valódi ár)}}{\text{n}} \times 2 + \$10 = \text{bemondott ár}$$

A hasraütésszerűen kitalált szám (n) a kulcs. Ugyanezt az algoritmust lehet minden alkalommal használni, amikor a mama látogatóba jön. Az anya nem tudja meghatározni a bútorok valódi árának elfedésére használt képletet, amíg nem ismeri a számításban alkalmazott kulcsot.

Ha a férfi egy székkel vagy asztallal tér haza, és meglátja az anyját a kertben, titokban jelez egy számot a barátnőjének (lásd a 23.1. ábrát). Amikor az anyja megkérdezi a bútor árát, a férfi végrehajtja az algoritmust, a barátnőjének kulcsként átadott számot használva. Ha a kulcs például 3, és a szék 600 dollárba került, a számítás így fest:

$$\frac{\$600}{3} \times 2 + \$10 = \$410$$

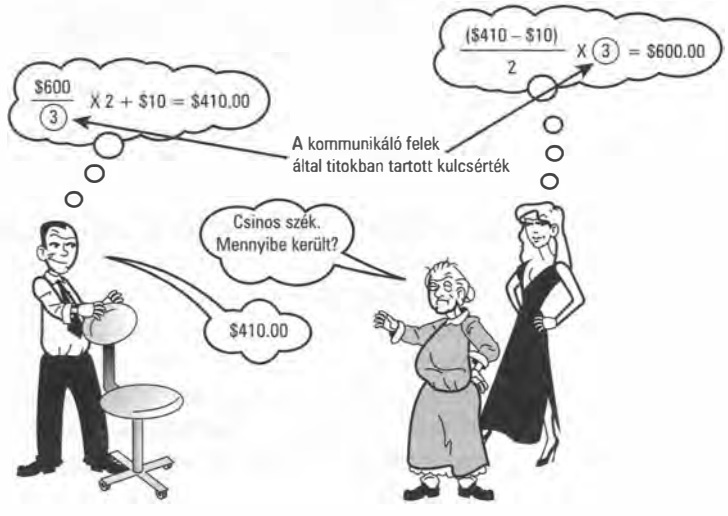
A barátnő, aki ismeri a titkos számítást, tudja, hogy az algoritmust fordítva végrehajtva kaphatja meg a valódi árat:

$$\frac{(\$410 - \$10)}{2} \times 3 = \$600$$

Ez az egyszerű példa, amelyet csak az algoritmus és a kulcs közötti különbség szemléltetésére mutattunk be, nem árulkodik a számítógépes titkosítási eljárások valódi bonyolultságáról. Azt sem szabad elfelejtenünk, hogy egy érték megváltoztatása nem ugyanaz, mint az adatok olvashatatlanná tétele. A számítógépek bináris világában ugyanakkor a határvonal kevésbé éles, mint amilyennek tűnhet.

A számítógép számára *minden* adat bináris adatbiteket jelent, amelyek egyeseket és nullákat ábrázolnak, ezért matematikai műveletek végezhetők rajtuk. Bármely eljárás, amely adatbitek egy sorozatát eltérő adatbit-sorozattá alakítja, elrejti az információ természetét. A lényeg az, hogy a fogadónak rendelkeznie kell valamilyen módszerrel, amellyel visszafejtheti a titkosított adatokat, hogy felfedje az eredeti információt, a titkosítási eljárásnak pedig valamilyen megosztott közös értéket (egy kulcsot) kell alkalmaznia, ami nélkül a visszafejtés lehetetlenné válik.

Szinte minden biztonságos hálózati eljárás alapját a titkosítás jelenti. A biztonságos rendszerek jelszavakat, bejelentkezési eljárásokat, sőt néha teljes kommunikációs munkameneteket titkosítanak. A titkosítási eljárás általában láthatatlan a felhasználó számára, bár a titkosítást végző alkalmazásokat és összetevőket többnyire szándékosan hívja meg a fejlesztő vagy a hálózati rendszergazda.



23.1. ábra
Rendkívül egyszerű algoritmus a kommunikáció álcázására

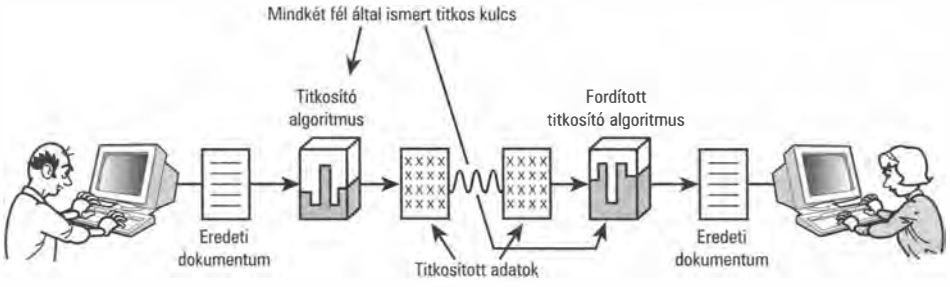
23

Szimmetrikus (hagyományos) titkosítás

A szimmetrikus titkosítást néha *hagyományos titkosításnak* is hívják, mert régebbi az újabban kifejlesztett aszimmetrikus eljárásoknál. A szimmetrikus titkosítás ma is a titkosítás legelterjedtebb formája, bár a nyilvános kulcsú aszimmetrikus titkosítás (amellyel az óra későbbi részében foglalkozunk) az utóbbi időben jelentős figyelmet kap.

A szimmetrikus titkosítást azért hívják szimmetrikusnak, mert a visszafejtés folyamata pontosan a fordítottja a titkosításének. A szimmetrikus titkosítási-visszafejtési eljárás menetét a 23.2. ábrán láthatjuk. Az eljárás lépései a következők:

1. A küldő és a fogadó számítógép egyaránt kap egy titkos kulcsot.
2. A küldő számítógép egy előre megbeszélt titkosító algoritmussal és a titkos kulccsal titkosítja az adatokat.
3. A célszámítógép megkapja a titkosított (olvashatatlan) szöveget.
4. A fogadó számítógép egy visszafejtő algoritmussal – ami pontosan a fordítottja a 2. lépésben alkalmazott titkosító algoritmusnak –, valamint a titkos kulccsal visszafejti az adatokat.



23.2. ábra

A szimmetrikus titkosítás folyamata

Az előző részben szereplő példában a bútormániás férfi és a barátnője egy szimmetrikus algoritmust használt a szék valódi értékének elrejtésére. A fogadó fél visszafelé halad az eredeti algoritmuson, ugyanazt a titkos kulcsot használva, mint amivel az adatokat eredetileg titkosították.

Felmerülhet bennünk a kérdés, hogy egyáltalán hogyan lehetséges olyan titkosítási módszert alkalmazni, ami *nem* az eredeti kulcsot használja a megfordított algoritmussal az adatok visszafejtéséhez. Ez a kérdés érthető, ha figyelembe vesszük, hogy a titkosítás több évszázados, a görög és római időkbe visszanyúló történetében senki nem próbálta másképp egészen az 1970-es évekig. Az óra későbbi részében azonban megismerkedünk az aszimmetrikus titkosítással is.

A szimmetrikus titkosítás rendkívül biztonságos, ha körültekintően hajtják végre. Bármely (szimmetrikus vagy aszimmetrikus) titkosítási séma biztonsága nagyrészt az alábbiakon múlik:

- A titkosító algoritmus erőssége
- A kulcs(ok) erőssége
- A kulcs(ok) titkossága

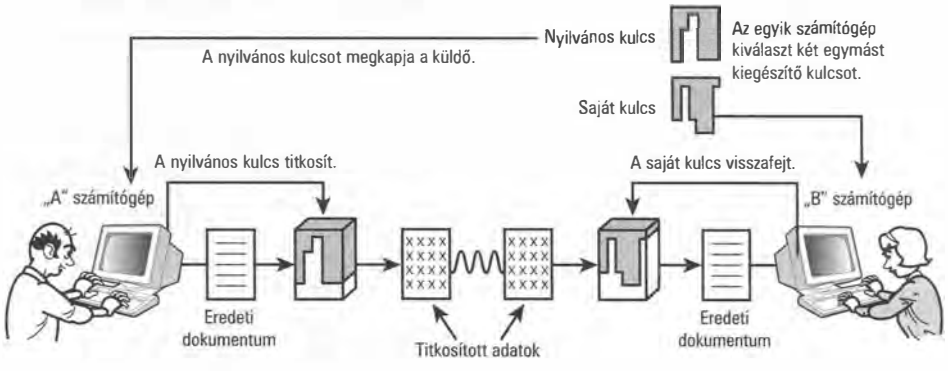
Egy 128 bites kulcsot használó titkosító algoritmus feltörése teljességgel lehetetlennek tűnhet, pedig megoldható. Az Interneten szabadon hozzáférhető kulcsfeltörő segédprogramokat találhatunk, és az egykor feltörhetetlennek vélt 128 bites titkosító algoritmusok egy része ma már nem számít biztonságosnak. A titkosított adatok megfejtését lehetővé tevő másik módszer a kulcs ellopása, ezért a szoftvernek valamilyen biztonságos módot kell nyújtania a kulcs továbbítására a fogadó számítógépnek. Különböző kulcstovábbító rendszerek léteznek, és néhányról szót is ejtünk az óra későbbi részében. A szimmetrikus titkosítás esetében minden a titkos kulcson áll vagy bukik. Ha elfogjuk a kulcsot, minden a kezünkbe kerül. A legtöbb rendszer ezért a kulcs időszakos megújítását követeli meg.

A kommunikáló számítógéppár által használt egyedi kulcs előállítása történhet munkamenetenként vagy egy adott idő letelte után. A kulcsmegújítás növeli a hálózaton áthaladó kulcsok számát, ami még fontosabbá teszi a kulcsok hatékony védelmét.

Több széles körben használatos titkosító algoritmus alapul szimmetrikus titkosításon. A *DES* (Data Encryption Standard, adattitkosítási szabvány) valaha népszerű választás volt, de az általa használt 56 bites kulcs ma már túl rövidnek számít. A mai titkosítási eljárások gyakran változtatható kulcshosszt engednek meg. A DES egyik leszármazottja, az *AES* (Advanced Encryption Standard, fejlett titkosítási szabvány) a 128, 192 és 256 bites kulcsokat támogatja, míg a Blowfish szimmetrikus algoritmus akár 448 bites kulcshosszt is megenged.

Aszimmetrikus (nyilvános kulcsú) titkosítás

Egy az utóbbi 30 évben fejlődésnek indult másik titkosítási módszer választ ad a kulcskiosztás egyes problémáira, amelyek a szimmetrikus titkosítás velejárói. Az aszimmetrikus titkosítást azért nevezik *aszimmetrikusnak*, mert az adatok titkosításához használt kulcs különbözik az adatok visszafejtéséhez használt kulcstól. Az eljárás menetét a 23.3. ábra szemlélteti.



23.3. ábra

Az aszimmetrikus titkosítás folyamata

Az aszimmetrikus titkosítást gyakran azonosítják a nyilvános kulcsú titkosítás módszerével. A nyilvános kulcsú titkosítás két kulcsot használ, amelyek közül az egyiket (az úgynevezett *saját kulcsot*) egyetlen számítógép tárolja, biztonságosan. A másik kulcsot (a *nyilvános kulcsot*) mindazok a számítógépek megkapják, amelyek adatokat akarnak küldeni a saját kulcsot tároló számítógépnek. Az eljárást a 23.3. ábrán láthatjuk, a lépései pedig a következők:

1. Az „A" számítógép megkísérel kapcsolatot létesíteni a „B" számítógéppel.
2. A „B" számítógépen található titkosító szoftver előállít egy saját és egy nyilvános kulcsot. A saját kulcsot „B" senkivel nem osztja meg, a nyilvános kulcsot viszont továbbítja „A"-nak.
3. Az „A" számítógép a „B" számítógéptől kapott nyilvános kulccsal titkosítja az adatokat, és elküldi azokat, a „B"-től kapott nyilvános kulcsot pedig elraktározza, ha később szükség lenne rá.
4. „B" megkapja az adatokat, és visszafejti azokat a saját kulcsával.

A nyilvános kulcsot alkalmazó módszerek egyik fontos jellemzője, hogy a nyilvános kulccsal végrehajtott titkosítás egyirányú. A nyilvános kulcs felhasználható az adatok titkosítására, de a rejtjelezés után csak a saját kulcs képes visszafejteni az adatokat. Ha valaki lehallgatja a hálózatot, és elfogja a nyilvános kulcsot, akkor sem tudja elolvasni az azzal titkosított üzeneteket.

Lehet persze azzal érvelni, hogy bár a lehallgató, aki elfogja a nyilvános kulcsot, nem tudja elolvasni az „A" számítógépről küldött adatokat, kiadhatja magát „A"-nak, új adatokat titkosíthat, és elküldheti azokat „B"-nek. Így annak ellenére, hogy a nyilvános kulcsú titkosítás bizalmasságot biztosít, nem feltétlenül nyújt hitelességet is. Mindazonáltal számos módszer létezik arra, hogy a titkosított adatok közé hitelesítő információkat csomagoljunk, hogy az adatok visszafejtésekor a „B" számítógép meggyőződhessen róla, hogy az adatok valóban az „A" számítógépről érkeztek (lásd az óra későbbi, *Digitális aláírások* és *Digitális tanúsítványok* című részeit).

A nyilvános kulcsú titkosítási módszereket széles körben használják védett internetes tranzakciókhoz. Az óra későbbi részében megismerkedünk majd a nyilvánoskulcs-tanúsítványokkal, amelyeket az olyan TCP/IP-biztonsági protokollokhoz használnak, mint a Secure Sockets Layer vagy az IP Security.

Digitális aláírások

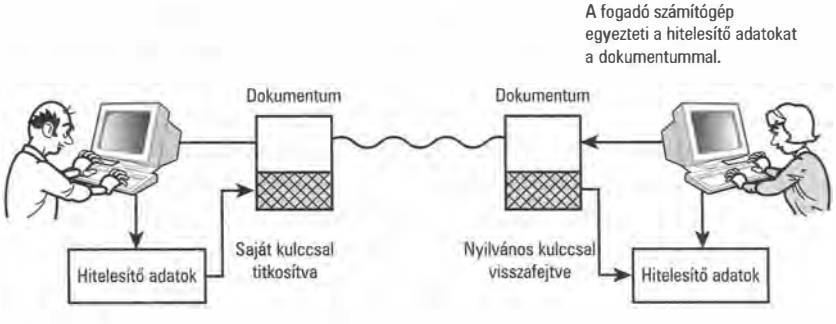
Időnként fontos, hogy biztosítsuk egy üzenet hitelességét, még ha nem is lényeges, hogy az üzenet tartalma bizalmas legyen. Egy tőzsdeügynök például kaphat egy ehhez hasonló üzenetet:

```
Adjon el 20 részvényt a Microsoft-csomagomból.
-Bennie
```

Húsz részvény eladása valószínűleg megszokott esemény az adott befektetőtől, ezért lehet, hogy sem a befektető, sem a tőzsdeügynök nem törődik azzal, hogy a tranzakció teljesen védett-e a lehallgatással szemben. Azt viszont kiemelt fontosságúnak tarthatják, hogy meggyőződjenek róla, hogy az eladásra utasító üzenetet valóban Bennie adta fel, és nem valaki más, aki Bennie-nek adja ki magát.

A digitális aláírás jó módszer arra, hogy biztosítsuk, hogy az adatok valóban abból a forrásból érkeznek, ahonnan állítják, és az adatokat nem változtatták meg valahol a kézbesítési útvonalon. A digitális aláírás egy titkosított adatblokk, amelyet egy üzenetbe ágyaznak. A titkosított adatblokkot néha *hitelesítőnek* is hívják. A digitális aláírások általában megfordítva alkalmazzák a nyilvános kulcsú titkosítás eljárását (lásd a 23.4. ábrát):

1. A „B" számítógép egy digitális aláírással ellátott dokumentumot akar küldeni az „A" számítógépnek. „B" létrehoz egy apró adatszakaszt a dokumentum tartalmának megerősítéséhez szükséges információkkal, más szavakkal, valamilyen matematikai számítást hajt végre a dokumentumot alkotó biteken, hogy levezessen egy értéket. Ez a hitelesítő szakasz az üzenet hitelességének megerősítéséhez más hasznos információkat is tartalmazhat, például egy időbélyeget vagy más jellemzőket, amelyek a hitelesítőt az üzenethez kapcsolják.
2. A „B" számítógép egy saját kulccsal titkosítja a hitelesítőt. (Vegyük észre, hogy ez a fordítottja az előző részben leírt nyilvános kulcsú titkosítási eljárásnak, ahol a saját kulcs az adatok visszafejtésére szolgált.) A hitelesítőt ezt követően a dokumentumhoz csatolja, és a dokumentumot elküldi az „A" számítógépnek.
3. Az „A" számítógép megkapja az adatokat, és visszafejti a hitelesítőt a „B" számítógép nyilvános kulcsával. A hitelesítőben található információk segítségével „A" meggyőződik róla, hogy az adatokat nem változtatták meg átvitel közben. Az a tény, hogy az adatokat vissza lehet fejteni a „B" számítógép nyilvános kulcsával, bizonyítja, hogy „B" saját kulcsával titkosították azokat, ami megerősíti, hogy az adatok valóban a „B" számítógépről származnak.



23.4. ábra
A digitális aláírás folyamata

A digitális aláírás tehát megerősíti, hogy az adatokat nem módosították, és azok valóban a feltételezett forrásból származnak. További (nem túl erős) biztonsági intézkedésként azt is meg lehet tenni, hogy nem csak a hitelesítőt, hanem a teljes üzenetet titkosítják a „B" számítógép saját kulcsával.

A saját kulccsal történő titkosítás és a nyilvános kulccsal végrehajtott visszafejtés ugyanakkor nem igazán nyújt bizalmasságot, mivel a visszafejtéshez használt nyilvános kulcs átadása az Interneten keresztül történik, így a titkossága nem feltétlenül biztosítható. Ha valaki lehallgatja a hálózatot, és megszerzi a nyilvános kulcsot, visszafejtheti a titkosított hitelesítőt – ugyanakkor új hitelesítőt nem tud titkosítani, és ezért nem adhatja ki magát a „B" számítógépnek.

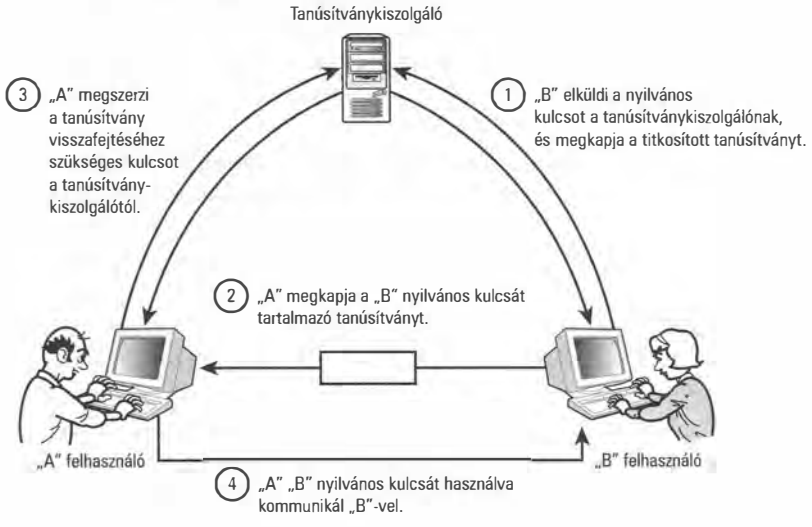
Digitális tanúsítványok

Az a nagy terv, hogy a nyilvános kulcsot bárki megkaphatja, aki kéri, érdekes megoldás, de megvannak a maga korlátai. Az a helyzet, hogy támadóknak így is van lehetőségük rá, hogy visszaéljenek a nyilvános kulcccsal: képesek lehetnek visszafejteni a digitális aláírásokat (lásd az előző részt), sőt akár a felhasználó saját kulcsával titkosított jelszavakat is elolvashatják. Ezért biztonságosabb működtetni valamilyen biztonsági rendszert, amely szabályozza, ki férhet hozzá a nyilvános kulcsokhoz.

A probléma egyik megoldását az úgynevezett digitális tanúsítványok jelentik. A digitális tanúsítvány lényegében a nyilvános kulcs egy titkosított másolata. A tanúsítási eljárást a 23.5. ábrán láthatjuk. Az eljáráshoz egy külső *tanúsítványkiszolgálóra* van szükség, amely biztonságos kapcsolatban áll mindkét kommunikálni kívánó féllel. A tanúsítványkiszolgálót hitelesítő hatóságnak (CA, Certificate Authority) is hívják.

Az Interneten több cég is nyújt tanúsítási szolgáltatást; az egyik legfontosabb hitelesítő hatóság a VeriSign Corporation. Egyes nagy szervezetek saját tanúsítási szolgáltatást üzemeltetnek, és a tanúsítás (hitelesítés) folyamata más és más lehet. Az eljárás nagy vonalakban az alábbi lépésekből áll:

1. „B" felhasználó egy biztonságos kapcsolaton keresztül másolatot küld a nyilvános kulcsáról a tanúsítványkiszolgálónak.
2. A tanúsítványkiszolgáló egy másik kulccsal titkosítja a „B" felhasználó nyilvános kulcsát (a felhasználó egyéb tulajdonságaival együtt). Ezt az új titkosított csomagot hívják tanúsítványnak. A tanúsítvány a tanúsítványkiszolgáló digitális aláírását is tartalmazza.
3. A tanúsítványkiszolgáló átadja a tanúsítványt a „B" felhasználónak.
4. „A" felhasználónak szüksége van „B" nyilvános kulcsára, ezért az „A" számítógép elkéri a „B" felhasználó tanúsítványának egy másolatát.
5. Az „A" számítógép egy biztonságos csatornán keresztül megkapja a tanúsítvány titkosításához használt kulcs másolatát a tanúsítványkiszolgálótól.
6. Az „A" számítógép visszafejti a tanúsítványt a tanúsítványkiszolgálótól kapott kulccsal, és kinyeri belőle a „B" felhasználó nyilvános kulcsát. Az „A" számítógép a tanúsítványkiszolgáló digitális aláírását is ellenőrzi (lásd a 2. lépést), hogy meggyőződjön róla, hogy a tanúsítvány hiteles.



23.5. ábra

Hitelesítés digitális tanúsítványokkal

A tanúsítási eljárás leginkább ismert szabványa az X.509, amelyet több RFC-dokumentum ír le. Az X.509 szabvány 3-as változatát az RFC 2459 tartalmazza.

A digitális tanúsítási eljárást arra találták ki, hogy egy felhasználói közösséget szolgáljon ki. Ahogy kitalálhatjuk, az eljárás biztonságossága a tanúsítványkiszolgálóval folytatott kommunikációhoz szükséges kulcsok biztonságos kiosztásán múlik. Ez persze úgy tűnhet, mintha csak továbbhárítanánk a problémát (mert a biztonságos kommunikációt a távoli állomással azáltal garantáljuk, hogy biztonságos kommunikációt tételezünk fel a tanúsítványkiszolgálóval). Az a tény azonban, hogy a védett kommunikációs csatorna egyetlen tanúsítványkiszolgálóra korlátozódik (és nem a közösségen belüli minden lehetséges állomásra), sokkal könnyebbé teszi a biztonságos adatcseréhez szükséges további óvintézkedések megtételét.

Az órában korábban leírt tanúsítási eljárás feltételezi, hogy az „A" számítógéphez rendelt tanúsítványkiszolgáló megegyezik azzal a kiszolgálóval, amelyik a „B" felhasználó számára nyújt tanúsítványokat. A tanúsítás valójában több tanúsítványkiszolgálót is igényelhet, amelyek szétszórtan helyezkednek el egy nagy hálózaton. Ilyen esetben az eljárás más tanúsítványkiszolgálókkal való üzenet- és tanúsítványváltások sorozatából állhat, amíg elérünk a „B" felhasználó tanúsítványát kibocsátó kiszolgálóig. Ahogy az RFC 2459 kimondja, „általában több tanúsítvány láncolatára lehet szükség, amelybe beletartozik a nyilvános kulcs tulajdonosának (a végfelhasználónak) a tanúsítványa, amelyet aláírt egy hitelesítő hatóság, valamint tanúsítványkiszolgálók nulla vagy több további tanúsítványa, amelyeket más hitelesítő hatóságok láttak el aláírással. Az ilyen láncolatokra, amelyeket *tanúsítási útvonalnak* hívnak, azért van szükség, mert

a nyilvános kulcsok felhasználói kezdetben csak korlátozott számú hiteles nyilvános kulccsal rendelkeznek." Szerencsére a titkosítással kapcsolatos részletek többségéhez hasonlóan a tanúsítási eljárásról is a szoftver gondoskodik, tehát nincs szükség a felhasználó közvetlen beavatkozására.

Az olyan TCP/IP-biztonsági protokollokban, mint a fejezet későbbi részében terítékre kerülő Secure Sockets Layer vagy az IP Security, az X.509 szabvány szerinti tanúsítási eljárást alkalmazzák.

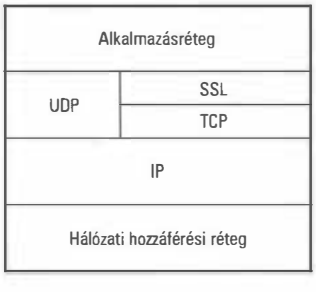
A TCP/IP biztosítása

Az utóbbi években a gyártók igyekeztek bővíteni a TCP/IP-megvalósításaikat, hogy beléjük építsék a fejezetben eddig bemutatott biztonsági és titkosítási eljárásokat. A következőkben azt vizsgáljuk meg, hogyan épülnek be a titkosítási eljárások két internetes biztonsági protokollrendszerbe: az SSL/TLS-be és az IPSec-be.

Más nyilvános biztonsági protokollok is fejlesztés alatt állnak, és a biztonsági programok gyártói közül néhányan saját rendszereket is fejlesztenek. A következőkben ezért csak általános képet szeretnénk adni arról, hogy milyen megoldások szükségesek ahhoz, hogy egy valódi hálózat működését titkosítással védjük.

Az SSL és a TLS

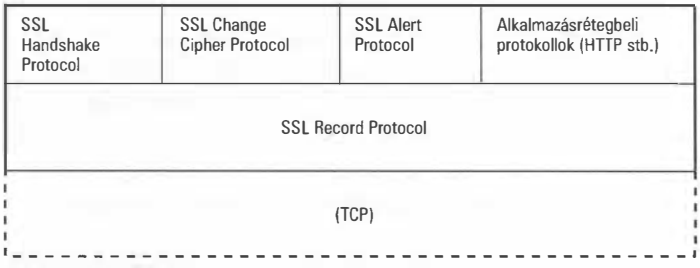
Az SSL (Secure Sockets Layer, biztonságos csatolóréteg) TCP/IP-biztonsági protokollok gyűjteménye, amelyet a Netscape fejlesztett ki a biztonságos webes kommunikációhoz. Az SSL célja, hogy egy biztonsági réteget nyújtson a szállítási (átviteli) réteg (lásd a 6. órát) csatolói és a hálózatot a csatolókon keresztül elérő alkalmazások között. Az SSL elhelyezkedését a TCP/IP-protokollveremben a 23.6. ábra szemlélteti. Az elv az, hogy amikor az SSL aktív, az olyan hálózati szolgáltatásokat, mint az FTP vagy a HTTP, a biztonságos SSL-protokollok védik a támadástól. A TLS (Transport Layer Security, biztonságos szállítási réteg) az RFC 2246-ban leírt protokollszabvány, amely az SSL 3.0-n alapul. A TLS-t gyakran tekintik az SSL utódjának. Az alábbiakban az SSL-t ismertetjük röviden – a TLS protokoll hasonlóan működik.



23.6. ábra
TCP/IP-verem SSL-lel

Ha közelebbről megnézzük az SSL réteget, két alréteget vehetünk észre (lásd a 23.7. ábrát). Az SSL Record Protocol (SSL rekordprotokoll) jelenti a szabványos alapot a TCP eléréséhez. A rekordprotokoll felett az SSL-hez kapcsolódó protokollok csoportját láthatjuk, amelyek mindegyike egy-egy szolgáltatást biztosít:

- **SSL Handshake Protocol (kézfogási protokoll)** – A TCP eléréséhez használt alapprotokoll.
- **SSL Change Cipher Spec Protocol (titkosító eljárást módosító protokoll)** – A titkosító eljárás beállításainak módosítását támogatja.
- **SSL Alert Protocol (riasztási protokoll)** – Riasztásokat küld.



23.7. ábra
SSL-alrétegek

Az SSL-képes szolgáltatások közvetlenül az SSL Record Protocolon keresztül működnek. Miután a kapcsolat létrejött, az SSL Record Protocol gondoskodik a munkamenet bizalmasságának és épségének biztosításához szükséges titkosításról és hitelesítésről.

Mint más protokollbiztonsági eljárások esetében, a kihívást itt is a résztvevők személyazonosságának ellenőrzése és az átvitt adatok titkosításához és visszafejtéséhez használt kulcsok biztonságos kicserélése jelenti. Az SSL nyilvános kulcsú titkosítást alkalmaz, és támogatja a digitális tanúsítványokat (lásd a fejezet korábbi részében). A kapcsolatot az SSL Handshake Protocol hozza létre, és ez a protokoll gondoskodik a kapcsolati beállításokról (köztük a titkosítási beállításokról) is.

Az SSL-t számos webhelyen használják arra, hogy biztonságos kapcsolatot létesítsenek pénzügyi információk és más érzékeny adatok cseréjéhez. A HTTP webprotokoll SSL-titkosítással ellátott változata a HTTPS. A legtöbb ma használatos böngésző képes SSL-kapcsolatot létrehozni a felhasználó közreműködése nélkül vagy minimális beavatkozással. Az egyik gondot az jelenti az SSL-lel kapcsolatban, hogy mivel az SSL a szállítási réteg felett működik, az ilyen kapcsolatot létesíteni kívánó alkalmazásoknak ismerniük kell az SSL-t. A következő részben egy másik TCP/IP-biztonsági rendszert (az IP Security-t) ismertetjük, amely egy mélyebb rétegben működik, és így elrejti a biztonsági rendszer részleteit az alkalmazás elől.

IPSec

Az IPSec (IP Security) egy másik biztonsági protokollrendszer, amelyet a TCP/IP-hálózatokon használnak. Az IPSec a TCP/IP-protokollvermen belül a szállítási réteg alatt működik. Mivel a biztonsági rendszer megvalósítása a szállítási réteg alatt helyezkedik el, a szállítási réteg felett tevékenykedő alkalmazásoknak nem kell tudniuk a biztonsági rendszerről. Az IPSec-et úgy tervezték, hogy támogassa a bizalmasságot, a hozzáférés-szabályozást, a hitelesítést és az adatépséget. Az IPSec a visszajátszáson alapuló támadások ellen is véd, amikor is a támadó kinyer egy csomagot az adatfolyamból, és később használja fel.

Az IPSec-et, amely lényegében az IP protokoll bővítményeiből áll, több RFC-dokumentum írja le, például az RFC 2401, 2402, 2406 és 2408. Ezek az RFC-k az IP Security bővítményeket mind az IPv4, mind az IPv6 számára meghatározzák. Az IPSec-et az IPv6 protokollrendszer szerkezetébe be is építették, míg az IPv4-ben bővítménynek számít, attól függetlenül, hogy az IPSec támogatását számos IPv4-megvalósítás tartalmazza.

Az IPSec a titkosításon alapuló biztonság előnyeit nyújtja bármely hálózati alkalmazásnak, függetlenül attól, hogy az adott alkalmazás ismeri-e a biztonsági rendszereket. Ugyanakkor a kommunikációban részt vevő mindkét számítógép protokollvermének támogatnia kell az IPSec-et. Mivel a magasabb szinten levő alkalmazások számára a biztonsági rendszer láthatatlan, az IPSec ideális biztonsági megoldás az olyan hálózati eszközök számára, mint az útválasztók vagy a tűzfalak. Az IPSec kétféle módban képes működni:

- A szállítási mód (transport mode) az IP-csomagok értékes tartalmát titkosítja, majd az értékes tartalmat egy normál IP-csomagba zárja a kézbesítéshez.
- A csatorna mód (tunnel mode) a teljes IP-csomagot titkosítja, és ebből a titkosított csomagból lesz egy másik, külső csomag értékes tartalma.

A csatorna módot arra használják, hogy biztonságos kommunikációs csatornát építsenek fel, amelyben a hálózattal kapcsolatos minden részletet elrejtenek. A hallgatózó támadó még a fejlécet sem tudja elolvasni, hogy megszerezze a forrás IP-címét. Az IPSec csatorna módját gyakran használják a VPN-temékek, amelyeknek a feladata az, hogy teljesen privát kommunikációs csatornát hozzanak létre egy nyilvános hálózaton keresztül.

Az IPSec többféle titkosító algoritmust és kulcskiosztási módszert alkalmaz. Az adatokat olyan hagyományos titkosító algoritmusokkal titkosítja, mint az AES, az RC5 vagy a Blowfish, a hitelesítéshez és a kulcskiosztáshoz pedig nyilvános kulcson alapuló eljárásokat használhat.

Virtuális magánhálózatok

A távoli eléréssel kapcsolatos problémákról sokszor szót ejtettünk ebben a könyvben. Ezek a problémák valójában fontos kérdést jelentettek a TCP/IP egész története során. Hogyan kapcsolhatunk össze olyan számítógépeket, amelyek nincsenek elég közel egymáshoz ahhoz, hogy helyi hálózatként, kábellel kössük össze őket? A rendszergazdák mindig is két fontosabb módszert alkalmaztak a távoli kapcsolatokhoz:

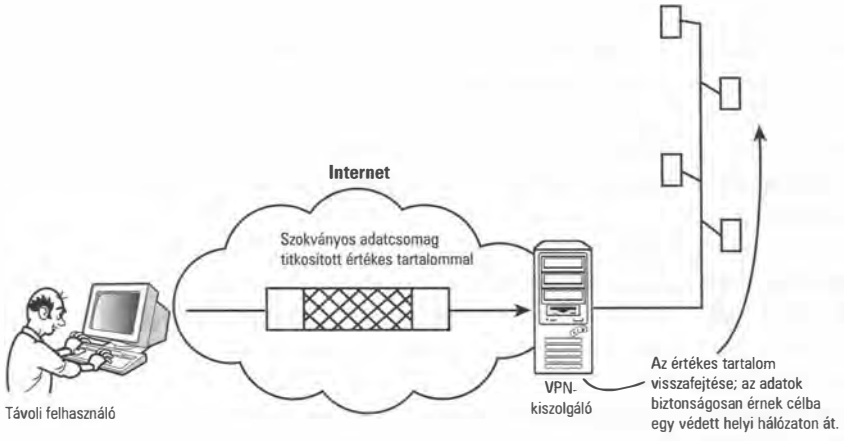
- **Betárcsázós kapcsolat** – A távoli felhasználó egy modemen keresztül kapcsolódik a betárcsázós kiszolgálóhoz, amely átjáróként működik a hálózat felé.
- **Nagy kiterjedésű hálózati kapcsolat (Wide Area Network, WAN)** – A két hálózatot egy kijelölt bérelt vonal kapcsolja össze egy telefontársaságon vagy internetszolgáltatón keresztül.

Mindkét módszernek vannak hátrányai is. A betárcsázós kapcsolatok hírhedten lassúak, és a sebességük függ a telefonvonal minőségétől. Néha a WAN-kapcsolatok is lehetnek lassúak, de ami fontosabb, hogy egy nagy kiterjedésű hálózat felépítése és fenntartása költséges, és a hálózat nem mobil, tehát a WAN-kapcsolat nem alkalmas arra, hogy kiszolgáljon egy távoli felhasználót, aki utazás közben a laptopját magával hurcolva változtatja a helyét.

Az egyik megoldás ezekre a problémákra, ha a nyílt Interneten keresztül csatlakozunk közvetlenül a távoli hálózatra. Ez a megoldás gyors és kényelmes, de az Internet annyira ellenséges és nem biztonságos környezet, hogy egyszerűen nem jöhet szóba, hacsak nem gondoskodunk valamilyen módszerrel a lehallgatás kivédéséről. A szakértők azon kezdtek töprengeni, hogy nem lehetne-e valahogy a titkosító eszközök segítségével magáncsatornát létrehozni egy nyilvános hálózaton keresztül, és ebből született meg a ma virtuális magánhálózatként (VPN, Virtual Private Network) ismert megoldás. A VPN ponttól pontig terjedő „csatornát" hoz létre a hálózaton keresztül, amelyen a szokványos TCP/IP-adatforgalom biztonságosan haladhat át.

Míg az IPSec (amelyről a fejezet korábbi részében beszéltünk) egy protokoll, amely a biztonságos hálózati kapcsolatokat támogatja, a VPN maga is kapcsolattípus. A VPN-alkalmazások olyan programok, amelyek távoli magánkapcsolatokat hoznak létre és tartanak fenn. Egyes VPN-eszközök az IPSec-et titkosításra használják, míg mások más SSL- vagy egyéb titkosítási eljárásokra támaszkodnak. A Microsoft-rendszerek a VPN-t korábban a *Point to Point Tunneling Protocolon* (pont-pont bújtatási protokoll, ebből származik a mai PPP modemprotokoll) keresztül biztosították; az újabb Microsoft-rendszerek azonban a VPN-munkamenetekhez már a *Layer 2 Tunneling Protocolt* (L2TP) alkalmazzák.

A fejezetben korábban ismertetett titkosítási eljárások nem működnének megfelelően, ha a kézbesítési lánc minden útválasztójának ismernie kellene a titkosító kulcsot. A titkosítást pont–pont kapcsolatokhoz szánták. Az elv az, hogy a VPN-ügyfélszoftver a távoli kiszolgálón kapcsolatot létesít egy VPN-kiszolgálóval, amely átjáróként működik a hálózat felé (lásd a 23.8. ábrát). A VPN-ügyfél és -kiszolgáló egyszerű, irányítható TCP/IP-adatcsomagokat cserél, amelyek normális esetben az Interneten haladnak keresztül. A VPN-kapcsolaton át elküldött értékes tartalom (az adatok) azonban valójában egy titkosított adatcsomag. A titkosított adatcsomagokat (amelyek a nyílt Interneten olvashatatlanok) a VPN-kiszolgálónak továbbított egyszerű, olvasható adatcsomagokba zárják. A VPN-kiszolgálószoftver ezt követően kinyeri a titkosított adatcsomagot, visszafejti azt a titkosító kulcs segítségével, és a benne foglalt adatokat továbbítja a rendeltetési helyére a védett hálózaton.



23.8. ábra

A virtuális magánhálózatok magáncsatornát biztosítanak egy nyilvános hálózaton keresztül

Bár lehetséges, hogy egy hallgatózó kibertolvaj elfogjon egy nem titkosított csomagot, ami a VPN-ügyfél és -kiszolgáló között halad, a hasznos információ teljes egészében a titkosított értékes tartalomban található, amelyet a lehallgató személy nem tud visszafejteni a szükséges kulcs nélkül.

A virtuális magánhálózatok elterjedésével ma már megszokott, hogy a felhasználók biztonságos, helyi hálózathoz hasonló kapcsolatokat létesítsenek távoli hálózatokkal az Interneten keresztül. A legtöbb rendszeren a VPN-kapcsolat létrehozásának és fenntartásának részleteit a szoftver kezeli, a felhasználónak csupán el kell indítania a VPN-alkalmazást, és meg kell adnia az azonosító adatait. Miután a kapcsolat létrejött, a felhasználó úgy használhatja a hálózatot, mintha az helyi kapcsolat lenne.

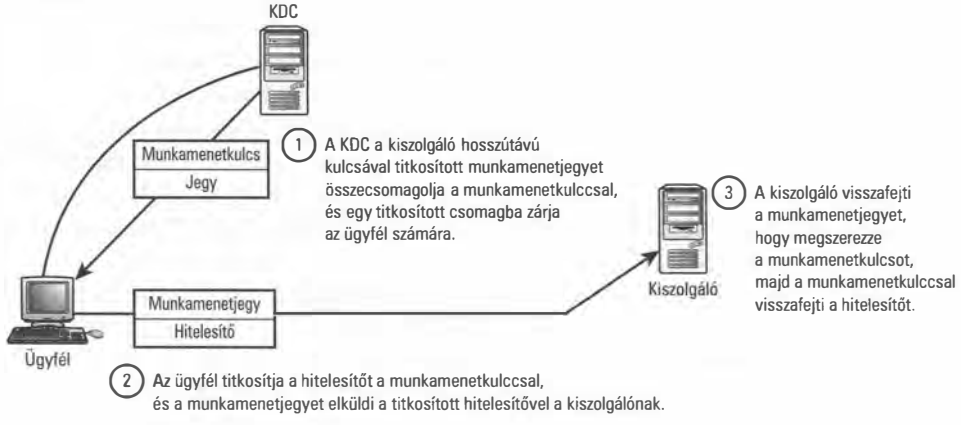
Kerberos

A Kerberos egy hálózati hitelesítési és hozzáférés-szabályozási rendszer, amelyet az ellenséges hálózatokon keresztüli biztonságos hozzáférés támogatására terveztek. A Kerberost az MIT-n fejlesztették ki az Athena projekt keretében. A Kerberos rendszert eredetileg Unix alapú rendszerekhez szánták, de azóta más környezetekre is átültették. A Microsoft is kínál egy Kerberos-változatot a Windows-hálózatokhoz.

Ahogy mostanra már bizonyára kitaláltuk, az ellenséges hálózatokon folytatott biztonságos kommunikáció kulcsát a titkosítás jelenti. Hosszabban kifejtve, a titkosító kulcsok biztonságát kell valamilyen módon biztosítanunk. A Kerberos módszeres eljárást nyújt a kulcsok kiosztására a kommunikáló állomásoknak, illetve a szolgáltatásokhoz hozzáférést kérelmező ügyfelek azonosítóinak ellenőrzésére.

A Kerberos rendszer egy kulcskiosztó központnak (KDC, Key Distribution Center) nevezett kiszolgálót használ a kulcskiosztás folyamatának kezelésére. A Kerberos hitelesítési eljárásnak három résztvevője van:

- **Az ügyfél** – A kiszolgálóhoz hozzáférést igénylő számítógép.
- **A kiszolgáló** – A hálózaton valamilyen szolgáltatást kínáló számítógép.
- **A kulcskiosztó központ (KDC)** – A hálózati kommunikációhoz kulcsok biztosítására kijelölt számítógép.



23.9. ábra

A Kerberos hitelesítési eljárás

A Kerberos hitelesítési eljárást a 23.9. ábra mutatja. Figyeljük meg, hogy az eljárás feltételezi, hogy a KDC már rendelkezik egy megosztott titkos kulccsal, amelyet az ügyféllel folytatott kommunikációhoz használhat, valamint egy másik megosztott titkos kulccsal, amely a kiszolgálóval való társalgásra szolgál. Ezeknek a kulcsoknak a segítségével egy

új munkamenetkulcsot titkosítanak, amelyet az ügyfél és a kiszolgáló az egymással való kapcsolattartásra használ. A KDC által az ügyfélnek és a kiszolgálónak küldött adatok titkosítására használt kulcsokat *hosszútávú kulcsnak* hívják. A hosszútávú kulcsot általában a KDC és a másik számítógép által megosztott kulcsból vezetik le. A hosszútávú ügyfélkulcs rendszerint a felhasználó bejelentkezési jelszavának kivonatértékéből (hash) származik, amelyet az ügyfél és a KDC egyaránt ismer.
Az eljárás menetét az alábbiakban ismertetjük. Olvasás közben ne feledjük, hogy a Kerberos általában hagyományos (szimmetrikus) titkosítást, és nem nyilvános kulcsú (aszimmetrikus) titkosítást alkalmaz; más szavakkal, az adatcsere mindkét végén ugyanazt a kulcsot használják:

1. Az ügyfél hozzá szeretne férni az „A" kiszolgáló valamelyik szolgáltatásához, ezért erre irányuló kérelmet küld a KDC-nek. (Egyes esetekben az ügyfél már átesett egy hitelesítési eljáráson, és kapott egy külön munkamenetkulcsot a KDC jegykibocsátó szolgáltatásával folytatott kommunikáció titkosításához.)
2. A KDC a következő lépéseket hajtja végre:
   a. Előállít egy munkamenetkulcsot, amely majd az ügyfél és az „A" kiszolgáló közötti kommunikáció titkosítására szolgál.
   b. Létrehoz egy munkamenetjegyet. A munkamenetjegy a 2a lépésben előállított munkamenetkulcs másolatát tartalmazza, valamint egy időbélyeget, és információkat a hozzáférést kérelmező ügyfélről, például az ügyfél biztonsági beállításait.
   c. Titkosítja a munkamenetjegyet az „A" kiszolgáló hosszútávú kulcsával.
   d. Összecsomagolja a titkosított munkamenetjegyet, a munkamenetkulcs egy másolatát és más válaszparamétereket az ügyfél számára, és az egész csomagot titkosítja az ügyfél kulcsával, majd elküldi a választ az ügyfélnek.
3. Az ügyfél megkapja a választ a KDC-től, és visszafejti azt. Így hozzájut az „A" kiszolgálóval való kommunikációhoz szükséges munkamenetkulcshoz, valamint a munkamenetjegyhez, amelyet a csomag a kiszolgáló hosszútávú kulcsával titkosítva tartalmaz. Az ügyfél nem tudja elolvasni a munkamenetjegyet, de azt tudja, hogy hitelesítésre át kell adnia azt a kiszolgálónak. Az ügyfél ehhez létrehoz egy hitelesítőt (egy karakterláncot, amely a hitelesítési paramétereket tartalmazza), és titkosítja azt a munkamenetkulccsal.
4. Az ügyfél hozzáférési kérelmet küld az „A" kiszolgálónak. A kérelem tartalmazza a munkamenetjegyet (a kiszolgáló hosszútávú kulcsával titkosítva), valamint a hitelesítőt (a munkamenetkulccsal titkosítva). A hitelesítőben többek között a felhasználó neve és hálózati címe, illetve egy időbélyeg található.
5. Az „A" kiszolgáló megkapja a kérelmet, a hosszútávú kulcsával visszafejti a munkamenetjegyet (lásd a 2c lépést), kinyeri belőle a munkamenetkulcsot, és azzal visszafejti a hitelesítőt. A kiszolgáló ez után meggyőződik róla, hogy a hitelesítőben található információk megegyeznek a munkamenetjegyben található információkkal. Ha igen, megadja a hozzáférést a szolgáltatáshoz.
6. Nem kötelező utolsó lépésként, amennyiben az ügyfél ellenőrizni szeretné az „A" kiszolgáló azonosítóit, az „A" kiszolgáló titkosít egy hitelesítőt a munkamenetkulccsal, és visszaadja ezt a hitelesítőt az ügyfélnek.

A Kerberos rendszer fokozatosan egyre népszerűbbé kezd válni, mint a hálózatokhoz egységes bejelentkezést nyújtó módszer. A Kerberos 4 a DES titkosítást használta, amelyet – ahogy ezen az órán már említettük – sok titkosítási szakértő ma már nem tart biztonságosnak. A Kerberos 5 (amelynek a leírása az RFC 1510-ben található) ezért már más titkosítási eljárások mellett az AES-t támogatja.

Ha már olvastunk leírást a Kerberosról, valószínűleg ismerjük a hagyományos történetet arról, hogy a Kerberos honnan kapta a nevét. A görög mitológiában Kerberos (vagy Cerberus) egy háromfejű kutya, aki az alvilág kapuját őrzi. A történet úgy szól, hogy a három fej a Kerberos hitelesítési eljárás három résztvevője (az ügyfél, a kiszolgáló és a KDC). A név azonban eredetileg másra utalt. *Network Security Essentials* című könyvében (Prentice Hall) William Stallings rámutat, hogy a hálózat kapuját a Kerberos rendszer eredetileg a hitelesítés, fiókkezelés és érvényességvizsgálat három fejével védte volna, de az utóbbi kettőt (fiókkezelés és érvényességvizsgálat) soha nem valósították meg. A biztonsággal foglalkozó közösség nyilván egyszerűbbnek találta másik hasonlatot keresni, mint átnevezni a protokollt egy megfelelő egyfejű kutya, például Lassie vagy Buck, a szánhúzó nevére.

Összefoglalás

Ezen az órán a TCP/IP-kommunikáció biztonságossá tételének néhány széles körben alkalmazott eljárását mutattuk be. Tanultunk a szimmetrikus és aszimmetrikus titkosításról, a digitális aláírásokról, valamint a digitális tanúsítványokról, és megismertük az olyan TCP/IP-biztonsági protokollrendszereket, mint az SSL és az IPSec, az órát pedig a Kerberos hitelesítés ismertetésével zártuk.

Kérdezz-felelek

K *Bob titkosított egy fájlt, hajlékonylemezre másolta, és mellé tette a fájl visszafejtéséhez szükséges kulcsot is. Bob titkosító programja szimmetrikus vagy aszimmetrikus titkosítást alkalmaz?*

V Bob titkosító programja szimmetrikus titkosítást alkalmaz. Ezt onnan tudhatjuk, hogy ugyanazt a kulcsot szándékozik a fájl visszafejtésére használni, mint amelyikkel titkosította azt. Nem csoda, ha furcsának találjuk, hogy Bob rátette a kulcsot a floppy-ra a titkosított fájllal együtt, mert ez valóban nem jó ötlet. Mi értelme titkosítani egy fájlt, ha mellékeljük hozzá a kulcsot? Így bárki, aki birtokába jut a fájlnak, a kulcsot is meg fogja találni.

K *Miért nem működik az SSL az UDP-vel?*

V Ahogy a 6. órán megtanultuk, az UDP a TCP-hez hasonlóan a szállítási réteg egyik protokollja, amely kapukat és csatolókat is biztosít a hálózat eléréséhez. Az SSL-nek ugyanakkor egy kapcsolaton keresztül kell működnie, az UDP pedig kapcsolat nélküli protokoll. Ez az oka annak, amiért az SSL-t csak a TCP-vel való működére tervezték.

K *Ellennek módot kell találnia arra, hogy több régi hálózati alkalmazást működésre bírjon egy Windows XP rendszerű számítógépen. Arra utasították, hogy ezekkel az ősrégi alkalmazásokkal gondoskodjon a kommunikáció bizalmasságáról. Az SSL-t vagy az IPSec-et érdemes használnia?*

F Az SSL a szállítási réteg felett működik, ezért az SSL-t használó alkalmazásoknak ismerniük kell az SSL felületet. Az IPSec ezzel szemben a verem alsóbb szintjén működik, így az alkalmazásoknak nem kell tudniuk róla. A fenti forgatókönyv azt valószínűsíti, hogy Ellen jobban jár, ha az IPSec-kel próbálkozik.

K *Mi történik, ha egy támadó valahogy rávesz egy Kerberos-ügyfelet, hogy rossz kiszolgálónak küldjön el egy munkamenetjegyet?*

F Semmi (legalábbis reméljük). A munkamenetjegyet a kiszolgáló hosszútávú kulcsa titkosítja, ezért amíg a támadó nem fér hozzá a kiszolgáló hosszútávú kulcsához, nem tudja feltörni a jegyet. Ha a támadó valahogy hozzájutna ehhez a hosszútávú kulcshoz, vissza tudná fejteni a jegyet, kinyerhetné belőle a munkamenetkulcsot, majd kiadhatná magát a kiszolgálónak.

Kulcsfogalmak

Ismételjük át az alábbi kulcsfogalmakat:

- **Advanced Encryption Standard (AES, fejlett titkosítási szabvány)** – A DES-en alapuló szimmetrikus titkosító algoritmus, amely a 128, 192 és 256 bites kulcshosszokat támogatja.
- **Aszimmetrikus titkosítás** – Titkosítási módszer, amely különböző kulcsokat használ a titkosításhoz és a visszafejtéshez.
- **Blowfish** – Szimmetrikus titkosító algoritmus, amely a legfeljebb 448 bites kulcshosszokat támogatja.
- **Hitelesítő hatóság (CA, Certification Authority)** – Tanúsítványok létrehozását és kiosztását felügyelő központi hatóság.
- **Data Encryption Standard (DES, adattitkosítási szabvány)** – Egykor népszerű, de a rövid, 56 bites kulcshossz miatt ma már nem biztonságosnak ítélt szimmetrikus titkosító algoritmus.
- **Digitális tanúsítvány** – Titkosított adatszerkezet, amelyet nyilvános kulcsok terjesztésére használnak.
- **Digitális aláírás** – Titkosított karakterlánc, amelyet a küldő személyazonosságának és az adatok épségének megerősítésére használnak.

- **Titkosítás** – Az a folyamat, amelynek során az adatokat szisztematikusan módosítják, hogy a jogosulatlan felhasználók számára olvashatatlanná tegyék azokat.
- **Titkosító algoritmus** – Matematikai képlet vagy eljárás adatok titkosításához.
- **Titkosító kulcs** – Titkosító algoritmuson belült használt (általában titokban tartott) érték, amelyet az adatok titkosításához és visszafejtéséhez használnak.
- **IPSec (IP Security)** – Biztonsági protokollrendszer, amely az IP protokoll bővítéseiből áll.
- **Kulcskiosztó központ (KDC, Key Distribution Center)** – A Kerberos-hálózatokon a kulcsok kiosztását kezelő kiszolgáló.
- **Kerberos** – Hálózati hitelesítési rendszer, amelyet szolgáltatások ellenséges hálózatokon történő biztonságos elérésére terveztek.
- **Saját kulcs** – Az aszimmetrikus titkosításban használt egyik kulcs, amelyet titokban tartanak, és nem visznek át a hálózaton.
- **Nyilvános kulcs** – Az aszimmetrikus titkosításban használt másik kulcs, amelyet átküldenek a hálózaton.
- **SSL (Secure Sockets Layer, biztonságos csatolóréteg)** – Biztonsági protokollrendszer, amelyet eredetileg a Netscape fejlesztett ki, és a TCP protokoll felett működik.
- **Szimmetrikus titkosítás** – Titkosítási módszer, amelynél a titkosító és a visszafejtő kulcs megegyezik.
- **X.509** – A digitális tanúsítási eljárást és a tanúsítványok formátumát leíró szabvány.

24. ÓRA

Egy TCP/IP-hálózat megvalósítása – egy rendszergazda hét napja

A fejezet tartalmából:

- A TCP/IP működés közben
- A hálózati rendszergazda feladatai

A könyv előző fejezeteiben a TCP/IP-hálózatok sok fontos elemével ismerkedtünk meg. Ezen az órán ezeket az elemeket valós – bár csupán elméleti – helyzetben láthatjuk, és az óra végeztével képesek leszünk leírni, hogyan működnek együtt egy TCP/IP-hálózat összetevői.

A Hypothetical Inc. rövid története

A Hypothetical Inc. („Képzelt vállalat") egy nagy és ormótlan vállalat, amely a semmiből indult, de ezt a semmit az idők során jelentősen megsokszorozta. 1987-es születése óta a Hypothetical Inc. a képzelt termékek gyártása és terjesztése mellett kötelezte el magát. A cég alapelve a következő:

> A legjobb képzelt termékeket elkészíteni és eladni bármikor és bármennyiért, amit csak a vásárló hajlandó rááldozni.

A gazdasági trendekkel lépést tartva a Hypothetical Inc. nemrégiben változásra szánta el magát. A cég stratégiájának középpontjában most már az áll, hogy a képzelt termékeket ne termékeknek, hanem szolgáltatásoknak tekintsék. Ez a látszólag jelentéktelen változtatás szigorú és szélsőséges intézkedéseket igényelt a megvalósítás tekintetében, amelyek azzal a súlyos következménnyel jártak, hogy az alkalmazottak munkamorálja mélyre süllyedt, és elharapózott az apró-cseprő alkatrészek szétlopkodása.

Az elégedetlenség okainak kivizsgálására felállítottak egy morálbizottságot, ami az elnökből, az alelnökből, az ügyvezető igazgatóból és az elnök unokaöcséből állt (ez utóbbi a postázóban dolgozik). Egyetértettek abban, hogy a cégnek ideje változtatnia azon a régi elvén, hogy nem használnak számítógépeket. (A „nincs számítógép" irányelv, amely még a képzelt termékek iparágának poshadt állóvizében is idejétmúltnak számított, természetes folyománya volt a vállalat hivatalos jelszavának: „Minden, amire szükségünk van az üzleti sikerhez, már a kőkorszakban is rendelkezésre állt.")

A bizottság tagjai, akik közül egyesek a közszférában tettek szert a tudásukra, azonnal megszavazták 1000 vegyes rendszerű számítógép (a nagy tételnek köszönhetően árengedményes) megvásárlását, azzal, hogy a szoftveres és hardveres eltéréseket majd később kiküszöbölik.

Elhelyezték az 1000 számítógépet a vállalat íróasztalain, pultjain, tárgyaló- és pihenőtermeiben, és összekötötték őket mindenféle kábellel, amit csak be tudtak dugni a különféle átalakítók segítségével. Nagy megdöbbenésükre a hálózat teljesítménye nem esett az elfogadhatóság határain belülre – valójában a nullával volt egyenlő, ezért keresni kezdtek valakit, aki képes elhárítani a hibát, vagy akire legalább rákenhetik az egész balhét.

Hét nap Maurice életéből

Maurice soha nem kételkedett benne, hogy egyszer munkát fog találni. Még totyogós korában átprogramozta a Candy Kinetic Sing-and-Stomp táncszőnyegét, hogy Dvorák Új világ szimfóniáját játssza, és azóta is egyedülálló képességekről tett tanúbizonyságot, ha a lehetetlen megvalósításáról volt szó a kibertérben. Arra azonban nem számított, hogy érettségi után ilyen hamar álláshoz jut – azt pedig végképp nem várta, hogy az a cég hívja be interjúra, akiknek az irodaházába csak azért tévedt be véletlenül, hogy igénybe vegye a mosdót. Mivel fiatal és merész volt, elfogadta a felkínált hálózati rendszergazdai állást a Hypothetical Inc.-nél, bár visszatekintve rá kellett volna jönnie, hogy előmenetelre innen nincs komoly lehetőség. Az interjún elmondta, hogy semmilyen tapasztalattal nem rendelkezik, de úgy tűnt, ez mit sem számít. Leendő munkaadói kifejtették, hogy a tapasztalat hiánya nekik megfelel, mert így alacsonyabb fizetést adhatnak. Ahelyett tehát, hogy ajtót mutattak volna neki, rögtön a kezébe nyomtak egy W-4-es űrlapot és egy tollat.

Kitűnő számítógépes könyvekből álló könyvtára sietett a segítségére, többek között a *Tanuljuk meg a TCP/IP használatát 24 óra alatt!*, amely közérthető, kerek bevezetést nyújtott a TCP/IP világába.

1. nap: Az első lépések

Amikor Maurice megérkezett a munkahelyére az első munkanapján, tudta, hogy az első feladata az kell legyen, hogy minden számítógépet a hálózatra kapcsoljon. A számítógépek gyors leltárba vétele néhány DOS és Windows, pár Linux, néhány Macintosh, és számos Unix rendszerű gépet fedett fel, valamint néhány olyat, amit még csak fel sem ismert. Mivel a hálózatot elvileg csatlakoztatnia kellett az Internethez (a bizottság munkamorál-javító intézkedései közül több is megkövetelte bizonyos meg nem nevezett rekreációs webhelyek látogatását), Maurice tudta, hogy a hálózaton a TCP/IP használatára lesz szükség, ezért gyorsan ellenőrizte, hogy fut-e a hálózat gépein a TCP/IP. A windowsos gépeken például az IPConfig segédprogrammal íratta ki a TCP/IP-paramétereket, a Unix és Linux rendszert futtatókon pedig az `ifconfig` segédprogramot használta erre a célra.

A legtöbb esetben úgy találta, hogy a TCP/IP valóban fut, de az IP-címek – nem kis meglepetésére – tökéletes összevisszaságot mutattak. A címeket láthatóan véletlenszerűen választották ki – nem volt két cím, amelyben hálózati azonosítóként felhasználható azonos számjegyek lettek volna. Mindegyik számítógép azt hitte, hogy önálló hálózaton található, és mivel egyik géphez sem rendeltek alapértelmezett átjárót, a kommunikáció mind a hálózaton belül, mind azon túl rendkívüli mértékben korlátozott volt. Maurice ekkor megkérdezte a felettesét (az unokaöcsöt, aki a postázóban dolgozott), hogy rendeltek-e internetes hálózati azonosítót a hálózathoz, mivel gyanította, hogy

a hálózathoz tartozik valamilyen előre kiosztott hálózati azonosító, hiszen a cég állandó internetkapcsolattal rendelkezett. Az unokaöcs semmiféle hálózati azonosítóról nem tudott.

Maurice megkérdezte tőle azt is, hogy a kereskedő, aki eladta nekik az 1000 számítógépet, beállította-e bármelyiket. Az unokaöcstől azt a választ kapta, hogy egyetlen számítógép beállítására került sor, mielőtt a partner hirtelen távozott volna a székházból a szerződés körüli vita miatt. Meg is mutatta Maurice-nak a kérdéses számítógépet: két kábel vezetett ki belőle – egy a céges hálózathoz, egy pedig az Internethez.

„Többotthonos rendszer" – mondta Maurice (az unokaöcs nem hatódott meg ettől). „Ezt a gépet átjáróként lehet használni" – magyarázta Maurice – „az üzeneteket az Internet felé tudja irányítani". Az unokaöcs türelmetlennek tűnt. Abban reménykedett, hogy gyorsan témát váltanak, és olyasmiről kezdenek beszélgetni, amiről nem Maurice, hanem ő tud többet.

A számítógép egy régi Windows NT-s rendszernek tűnt. Maurice eltűnődött rajta, hogy közölje-e az unokaöccsel, hogy még soha nem hallott olyasmiről, hogy valaki egy többotthonos Windows NT-gépet vállalati átjáróként használjon, és hogy sok szakértő ezt hívja „nagyon bizarr kiépítésnek", valamint hogy sokkal jobb megoldás lenne egy átjáró-útválasztót vásárolni. Mivel azonban ez volt az első napja, úgy döntött, hogy nem osztogat tanácsokat. Egy számítógép végtére is képes útválasztóként működni, amennyiben IP-továbbításra állítják be. Az átjáró számítógéptől egy Ethernet-kábel vezetett a hálózat többi részéhez. Maurice gyorsan kiadott egy `ipconfig` parancsot a gépen, hogy megszerezze az Ethernet-csatoló IP-címét. Volt egy olyan érzése, hogy a kereskedő helyes hálózati azonosítót állított be a gépen, mielőtt távozott volna. Az IP-cím `198.100.145.1` volt.

Maurice látta a pontozott jelölést használó cím első számából (198), hogy egy C osztályú hálózatról van szó. Egy C osztályú hálózatban az első három bájt adja meg a hálózati azonosítót. „A hálózati azonosító `198.100.145.0`." – közölte az unokaöccsel. Amíg ott tartózkodott, a TCP/IP-beállításokat is ellenőrizte, hogy meggyőződjön róla, hogy az IP-továbbítás be van kapcsolva.

Mauric-nak eszébe jutott, hogy a C osztályú címtérben rendelkezésre álló állomásazonosítókkal a hálózat csak 254 számítógépet támogat, de arra jutott, hogy ez valószínűleg nem számít, mert sok felhasználó úgysem használja a számítógépét, ezért valószínűtlen, hogy bármikor egyszerre 254 felhasználónál több éri el a hálózatot. Először beállította a morálbizottság tagjainak IP-címeit:

`198.100.145.10`	(elnök)
`198.100.145.3`	(alelnök)
`198.100.145.8`	(ügyvezető igazgató)
`198.100.145.5`	(unokaöcs)

Ezt követően megadta az összes egyéb lehetséges állomásazonosítót, valamint alapértelmezett átjáróként az átjáró számítógép címét (`198.100.145.1`), hogy az üzenetek és kérelmek a hálózaton túlra is továbbíthatók legyenek. Minden IP-cím esetében a C osztályú hálózatok szabványos alhálózati maszkját, a `255.255.255.0`-t használta.

Maurice ez után megvizsgálta a hálózat működését a `ping` segédprogrammal. Minden számítógépen beírta a `ping` parancsot a hálózat egy másik gépének címével együtt. A `198.100.145.155` címmel rendelkező számítógépen például a `ping 198.100.145.5` parancsot hajtotta végre, hogy meggyőződjön róla, hogy a gép felhasználója képes lesz-e kommunikálni az unokaöccsel. Ezenkívül, az ajánlott eljárást követve, mindig intézett egy visszhangkérést az alapértelmezett átjáróhoz is:

```
ping 198.100.145.1
```

Minden visszhangkérésre választ kapott a célszámítógéptől, ami biztosította arról, hogy a kapcsolat működik. Maurice úgy vélte, hogy a hálózat nagy lépést tett előre egyetlen nap alatt, és hogy könnyű és kifizetődő munkája lesz, de az utolsó számítógép, amelyet beállított, nem kapott választ a hálózat többi gépéhez intézett visszhangkéréseire. Alapos kutatás után Maurice észrevette, hogy a számítógép egy egészen más típusú fizikai hálózat részének tűnik. Valaki úgy próbálta meg az elavult, ismeretlen hálózati kártyát a hálózathoz csatlakoztatni, hogy beleerőszakolt egy 10BASE-2 Ethernet-kábelt a csatlakozójába. Mivel a kábel nem passzolt, az illető egy szöggel kötötte át az áramkört, és az egész tákolmányt betekerte szigetelőszalaggal, hogy úgy nézett ki, mintha az Apollo 13-on használták volna.

„Holnap." – sóhajtotta Maurice.

2. nap: Szakaszolás

Amikor Maurice másnap munkába ment, hozott magával valamit, amiről tudta, hogy szüksége lesz: útválasztókat. Bár korán érkezett, sok felhasználó türemetlenkedett vele: „Mi van ezzel a hálózattal?" – kérdezték – „Ez nagyon lassú!"

Maurice megmondta nekik, hogy még nem végzett. A hálózat működött ugyan, de az átvitelért versengő eszközök nagy száma lelassította. Ezen kívül egyes számítógépeket (az előző nap végén felfedezett számítógéphez hasonlóan) más hálózati felépítéshez állítottak be, és ezért nem tudtak közvetlenül kommunikálni a többi számítógéppel. Maurice a stratégiai pontokon felállított néhány útválasztót, hogy azok csökkentsék a hálózati forgalmat, és bevonják a más fizikai felépítéssel rendelkező hálózati elemeket. Természetesen találnia kellett egy olyan útválasztót, amelyik támogatta az elavult felépítést, de ez nem volt nehéz, mert Maurice jó kapcsolatokkal rendelkezett.

Maurice azt is tudta, hogy alhálózatokra kell osztania a hálózatot. Úgy döntött, hogy a C osztályú hálózatazonosító utáni utolsó nyolc bitet úgy osztja fel, hogy három bittel jelezhesse az alhálózat számát, a másik öttel pedig meghatározhassa az állomás azonosítóját az alhálózaton.

Az alhálózati maszk meghatározásához felírt egy 8 bites bináris számot (ami az utolsó oktettet jelképezte), az első három bit (az alhálózat bitjei) helyén egyesekkel, a maradék bitek (az állomás bitjei) helyén pedig nullákkal:

```
11100000
```

Az alhálózati maszk utolsó oktettje tehát 32 + 64 + 128, vagyis 224 lett, a teljes alhálózati maszk pedig `255.255.255.224`. Maurice ezt az új alhálózati maszkot hozzáadta az újonnan alhálózatokra osztott hálózathoz, és ennek megfelelően osztotta ki az IP-címeket – úgy hogy, hogy a három alhálózati bit egy adott szakasz minden számítógépén megegyezzen. Sok számítógépen az alapértelmezett átjáró értékeit is módosította, mert az eredeti átjáró többé már nem ugyanazon az alhálózaton helyezkedett el. Ehelyett a számítógépeken alapértelmezett átjáróként annak az útválasztókapunak az IP-címét adta meg, amelyhez az adott alhálózat csatlakozott.

3. nap: Dinamikus címek

A hálózat most már nagyszerűen működött, és Maurice eredményei tiszteletet vívtak ki a számára. Egyesek még arra is javaslatot tettek, hogy kerüljön be a morálbizottságba. Az unokaöcs mindazonáltal nem értett egyet ezzel. Szerinte Maurice nem érdemelte ki, hogy a morálbizottság – vagy bármilyen más bizottság – tagja legyen, mert mindezidáig nem teljesítette azt, amivel megbízták. A bizottság világosan megmondta, hogy a hálózatnak 1000 számítógépből kell állnia, Maurice viszont csak egy 254 gépből álló hálózatot adott nekik. „Hogyan várhatjuk el a munkamorál javulását, ha a morálbizottság utasításait figyelmen kívül hagyják?" – kérdezte.

Valójában a hálózat ebben a pillanatban 254 címnél kevesebbet tartalmazott, mert a második napon végrehajtott alhálózatokra bontás további, ki nem osztható címeket (a csupa nullából álló állomásazonosítót és a csupa egyesből álló adatszórási címet) hagyott minden alhálózaton. Egy alhálózaton belül a ténylegesen elérhető címek száma nem 2^n, hanem (2^n-2), ahol *n* az állomásazonosító bitek száma a címben. Maurice mindazonáltal nem látott rá okot, hogy felfedje ezt a tényt az unokaöcs előtt.

De hogyan tudna Maurice 1000 számítógépnek internetelérést biztosítani 254-nél kevesebb lehetséges állomásazonosítóval? Tudta, hogy a válasz az, hogy be kell állítania egy DHCP-kiszolgálót, amely ideiglenes IP-címeket kölcsönöz a felhasználóknak. „A DHCP elve az" – magyarázta – „hogy a felhasználók nem egyszerre használják

a számítógépüket". A DHCP-kiszolgáló számon tartja a rendelkezésre álló IP-címeket, és amikor egy számítógép elindul, és egy címet kér, hozzárendel egyet ideiglenesen. Amíg a felhasználók csak időnként használják a számítógépüket, 254 IP-címmel is ki lehet szolgálni 1000 számítógépet.

A címek szűkösségére egy másik megoldás egy hálózati címfordító (NAT, Network Address Translation) eszköz használata lett volna az internetkapcsolathoz. Ha NAT-eszközt használt volna, Maurice bármilyen címet kioszthatott volna a hálózaton, amit csak akar, függetlenül attól, hogy a címek a vállalat hivatalos címtartományába tartoznak-e. Egy olyan cégnél azonban, ahol házi gyártású ragacsot használnak még a szövegkihúzó korrektor helyett is, nem biztos, hogy egy új eszköz beszerzésére tett javaslat megértő fülekre talált volna. Az unokaöcs ráadásul határozottan territoriális viselkedést mutott az ócska Windows NT-s átjáró számítógép védelmében, és úgy tűnt, személyesen érinti annak sikere vagy kudarca.

A DHCP-kiszolgáló beállítása könnyű volt – legalábbis Maurice-nak, mert gondosan elolvasta a dokumentációt, és nem félt segítséget keresni a Weben sem. (Arról persze gondoskodnia kellett, hogy az útválasztók megfelelően legyenek beállítva, hogy átengedjék a DHPC-információkat.) A nehézséget az 1000 számítógép kézi beállítása jelentette a DHCP-kiszolgáló eléréséhez és a dinamikus IP-címek fogadásához. Ahhoz, hogy beállíthasson 1000 számítógépet egy nyolc órás munkanap alatt, Maurice-nak óránként 125 géppel kellett végeznie, vagyis percenként valamivel több, mint kettővel. Ez szinte bárkinek lehetetlen feladat lett volna – Maurice-t kivéve. Néhány embert ugyan le kellett csapnia útközben, de még időben végzett ahhoz, hogy elérje a du. 6:00-ás buszt.

4. nap: Tartománynév-feloldás

A következő napon Maurice rájött, hogy a hálózat sietős átállítása a dinamikus címkiosztásra feloldatlan konfliktusokat eredményezett. Ilyesmire sehol máshol nem került volna sor, csak a Hypothetical Inc.-nél, ott azonban valódi és sürgető problémát jelentett.

Az elnök magánbeszélgetésre hívta Maurice-t, és tájékoztatta, hogy ő, mint a vállalat legmagasabb rangú tisztségviselője, elvárja, hogy a számítógépe a számszerűen legalacsonyabb IP-címmel rendelkezzen. Maurice még soha nem találkozott ilyen kéréssel, és a dokumentációkban sem talált sehol ilyesmire utalást, de biztosította az elnököt, hogy a kérése nem jelenthet gondot. Egyszerűen úgy állítja majd be az elnök számítógépét, hogy a statikus `198.100.145.2` címet használja, és ezt a címet kizárja a DHCP-kiszolgáló által kioszthatő címtartományból. Maurice hozzátette, hogy reméli, hogy az elnök megérti, milyen fontos, hogy ne nyúljanak az internetes átjáróként üzemelő számítógép beállításaihoz, amelyet még a kereskedő állított üzembe, és amelynek egyedüliként alacsonyabb címe volt: `198.100.145.1`. (Valójában Maurice ezt a címet is átállíthatta volna valami magasabbra, de nem fűlt hozzá a foga.) Az elnök leszögezte, hogy nem

bánja, ha egy számítógépnek alacsonyabb IP-címe van, mint az övének, amíg az a gép nem egy alkalmazotté. Ő csupán azt akarja, hogy egyetlen személy se rendelkezzen az övénél alacsonyabb IP-címmel.

Maurice és az elnök megállapodása semmilyen módon nem hátráltatta volna a hálózat további fejlesztését, ha a felsőbb szintű vezetők nem követelték volna ugyancsak a helyüket a hiúságnak ezen a szomorú szamárlétráján. Az alelnöknek és az ügyvezető igazgatónak egyszerűen lehetett alacsony IP-címeket adni, de a középvezetők, akik egymással azonos szinten álltak, marakodni kezdtek azon, hogy melyikük számítógépe kapja a `198.100.145.33`-as, és melyiküké a `198.100.145.34`-es címet. Végül a menedzsment arra kényszerült, hogy visszavonuljon egy teniszpartira, ahol tisztázzák a nézeteltéréseket, és minden mérkőzést szeretettel vívnak meg.

Közben Maurice kidolgozott egy megoldást, amiről tudta, hogy el fogják fogadni. Felállított egy DNS-kiszolgálót, hogy cím helyett minden számítógépet névvel lehessen azonosítani, így minden vezető kiválaszthatja majd a saját gépe állomásnevét. A státuszt tehát nem az fogja jelezni, hogy kinek van a számszerűleg legalacsonyabb címe, hanem hogy kinek a számítógépe viseli a legötletesebb nevet. A középvezetők állomásnevei között például ilyenek szerepeltek:

- `Gregor`
- `wempy`
- `righteous_babe`
- `Raskolnikov`

A DNS-kiszolgáló jelenléte szintén közelebb vitte a vállalatot a hosszútávú célhoz: a teljes interneteléréshez. Más DNS-kiszolgálókkal fenntartott kapcsolatain keresztül a DNS-kiszolgáló teljes hozzáférést adott a cégnek az internetes állomásnevekhez, amilyeneket az Interneten az URL-ekben is használnak.

Maurice arra is szánt néhány percet, hogy igényeljen egy tartománynevet, hogy a cég egy napon képes legyen a képzelt termékeit a saját weboldalán árusítani a Világhálón.

5. nap: Tűzfalak

A hálózaton újonnan elért sikerek ellenére a cég munkamorálja még mindig alacsonyan állt. Az alkalmazottak egymás után mondtak fel, és úgy távoztak, mint a nézők egy rossz filmről a moziból. A távozó alkalmazottak közül sokan jól ismerték a hálózatot, és a vezetők aggódtak, hogy az elégedetlenek egyfajta bosszúként esetleg kibervandalizmusra adhatják a fejüket, ezért arra kérték Maurice-t, hogy valósítson meg egy olyan rendszert, amely megvédi a hálózati erőforrásokat, de a hálózat felhasználóinak a lehető legtelje-

sebb hozzáférést nyújtja a helyi hálózathoz és az Internethez egyaránt. Maurice megkérdezte, mennyi pénzt szánnak rá, és azt a választ kapta, hogy kivehet némi aprót a kávéfőző melletti malacperselyből.

Maurice eladott úgy ötvenet az 1000 számítógépből, és az értük kapott pénzt arra használta fel, hogy vásároljon egy kereskedelmi tűzfalrendszert, ami megvédi a hálózatot a külső támadásoktól. (Az ötven gép használaton kívül állt, és csak eltorlaszolta a folyosót. A takarítók legalább hatszor próbálták kidobni őket, hogy hozzáférjenek az ajtóhoz.) A tűzfal számos biztonsági szolgáltatást nyújtott, de az egyik legfontosabb az volt, hogy lehetővé tette Maurice-nak a TCP- és UDP-kapuk letiltását, ami megakadályozta, hogy a kívülről érkező felhasználók hozzáférjenek a hálózati szolgáltatásokhoz. Maurice minden nem létfontosságú kaput letiltott, de az FTP-elérést biztosító 21-es TCP-kaput nyitva hagyta, mert a Hypothetical Inc.-nél az információkat gyakran nagy, nyomtatható dokumentumokban tették közzé, amelyeknek a kézbesítésére az FTP ideális. Maurice gondosan úgy állította be a tűzfalat, hogy a 21-es kapun az FTP-hozzáférés csak egy jól védett FTP-kiszolgálóhoz való csatlakozásra legyen engedélyezett.

6. nap: Webszolgáltatások

A hálózat végre biztonságossá és jól szervezetté vált. A morálbizottság úgy döntött, hogy az új kommunikációs csatornát arra használja, hogy kémkedjen az alkalmazottak után, hogy felügyelhesse a termelékenységet. Meglepetésükre azt kellett megállapítaniuk, hogy valójában senki nem csinál semmit. Az új rendelések feldolgozása hatalmas lemaradást mutatott, mert a vállalatnak nem volt automatizált rendszere az új képzelt termékekre leadott rendelések rögzítésére, naplózására és feldolgozására. A látogatók elvileg FTP-n keresztül tölthették le az új képzelt termékeket. A kiszolgálón elhelyezett tájékoztató arra utasította a vásárlókat, hogy a pénzt a cég szekhelyére küldjék, ahol minden borítékot óvatosan felbontottak és megvizsgáltak az önkéntesek a dohányzóban.

Maurice elhelyezett egy webkiszolgálót a tűzfal előtt, és úgy állította be, hogy a vásárlók egy HTML-űrlapon keresztül adhassák fel a rendeléseiket. A webkiszolgáló elé egy másik tűzfalat helyezett, így egy demilitarizált zónát (DMZ) létrehozva a kiszolgáló és a többi internetkész számítógép számára. Ez után a belső hálózaton felállított egy újabb webkiszolgálót, és üzembe helyezett egy webszolgáltatásként működő alkalmazást azzal a feladattal, hogy gondoskodjon a rendelések feldolgozásáról és a raktárkészlet nyilvántartásáról. Az alkalmazottak asztali gépein egy apró ügyfélprogram kommunikált a kiszolgálóval, XML formátumú SOAP-üzenetek kicserélése révén. A külső webkiszolgáló, amely biztonságos csatornán keresztül kapcsolódott a belső kiszolgálóhoz, adta át a Webről érkező rendeléseket. A kiszolgálót egy háttéradatbázishoz kötötte, amely nyomon követte a vásárlói tranzakciókat, és egy hitelkártya-feldolgozó szolgáltatással létesített biztonságos kapcsolat gondoskodott az e-kereskedelem csodáiról a webhely látogatói számára.

A termelékenység gyorsan nőtt, így több idő jutott a kávészünetekre, és a vállalat hamarosan rájött, hogy nincs szüksége ennyi munkás kézre. A könyvelési részleg három dolgozója is majdnem lapátra került, de gondoskodtak a jövőjükről, azáltal, hogy kidolgoztak egy módszert az irodai bútorok felülvizsgálatára, amely biztosította, hogy az egymást követő asztalok és székek egymást követő sorozatszámmal rendelkezzenek.

Maurice-nak megengedték, hogy korán hazamenjen a munkából, de ő maradt, hogy üzembe állítson egy teljesítménynövelő fordított közvetítőrendszert a webhely számára.

7. nap: Aláírások és virtuális magánhálózatok

A webszolgáltatások új rendszere példátlan sikert hozott a Hypotheticals, Inc.-nek, és a vállalatot hirtelen elárasztották az új megrendelések. Mivel azonban a megrendelésfeldolgozó rendszerek mind automatizáltak voltak, a személyzet nem igazán vette észre ezt a szerencsés fordulatot, és a munkanap nagy részét továbbra is olyan értekezleteken töltötte, amelyeknek a célja további értekezletek tervezése volt. A siker azonban a versenytársak figyelmét nem kerülte el. Az egyik riválisa különösen érdeklődőnek mutatkozott. Bár ez a gyártó nem arról volt ismeretes, hogy kiváló minőségű vagy hatékony szolgáltatást nyújtana, a rendkívül alacsony költségeknek – a főhadiszállásukat egy elhagyott lakókocsiban ütötték fel – köszönhetően mégis sikerült talpon maradnia.

A versenytárs nem újítással válaszolt, hanem az egyetlen olyan módon, amit ismert – utánzással. Utánzásuk azonban túlment az eljárások puszta finomításán, és villámgyorsan áttévedt a jogsértő magatartás ködös terepére. Azt kezdték terjeszteni, hogy valójában *ők* a Hypotheticals, Inc., és a Hypotheticals, Inc. nevében kötöttek üzleteket. Mivel a tranzakciók távolról zajlottak, a megrendelők nem tudták ellenőrizni a szállító azonosságát.

Szerencsére Maurice rögtön készen állt a megoldással, amelyet – lévén hogy a vállalat többi alkalmazottja éppen kávészünetet tartott – minimális fennakadás mellett sikerült megvalósítania. A szükséges változtatásokat végrehajtva digitális aláírások rendszerét állította fel, amelyek a cég minden dokumentumának eredetét bizonyították.

Ennek az intézkedésnek a sikere jó ürügy volt egy ritka irodai ünnepségre, amelyen elismerték Maurice érdemeit, és repetát kínáltak neki a rágcsálnivalóból. A buli végeztével az ügyvezető igazgató zárt ajtók mögött beszélgetésre invitálta Maurice-t, és megkérdezte tőle, hogy a szövetségi törvény tiltja-e a nagy összegű sportfogadást az Interneten. Maurice azt válaszolta, hogy mivel nem jogász, és nem ismeri a szerencsejátékokat szabályozó törvény részleteit, nem tud felvilágosítással szolgálni.

Az igazgató ekkor arról érdeklődött, hogy Maurice nem ismer-e véletlenül valamilyen módszert arra, hogy az Interneten folytatott minden üzenetváltás szigorúan bizalmas legyen, hogy senki ne tudhassa, hogy mit mond, vagy hogy kivel társalog. Maurice azt

mondta neki, hogy erre a legjobb megoldás, amiről tudomása van, egy virtuális magánhálózat használata. A virtuális magánhálózat (VPN) egy titkosított magáncsatorna egy nyilvános vonal felett. A VPN olyan kapcsolatot biztosít, ami majdnem olyan bizalmas, mint egy pont–pont kapcsolat.

„Most rögtön szeretnék egy ilyet." – közölte vele az igazgató, majd gondolataiba merülve visszavonult az irodájába.

Összefoglalás

Ebben az órában egy képzeletbeli vállalat TCP/IP-hálózatát vettük górcső alá. Bepillantást nyerhettünk abba, hogy miért és hogyan valósítják meg a hálózati rendszergazdák az IP-címzést, az alhálózati maszkokat, a DNS-t, a DHCP-t és más szolgáltatásokat.

Ha kíváncsiak vagyunk a történet végére...

Pár nappal később szövetségi ügynökök keresték fel a cég főhadiszállását, és letartóztatták az ügyvezető igazgatót. Így megüresedett egy szék a morálbizottságban, amelyet az elnök nagylelkűen Maurice-nak ajánlott fel.

Kérdezz-felelek

K *Miért döntött Maurice úgy, hogy három bitet használ az alhálózati címhez?*

V Az alhálózati bitek ideális száma az alhálózatok számától és azok méretétől függ. Ha további biteket adunk az alhálózati számhoz, kevesebb bit marad az állomáscím számára. A fenti esetben Maurice a hálózat aktuális állapota alapján hozta meg a döntését. Egy hárombites maszk alhálózatonként 30 állomást tesz lehetővé.

K *Miért döntött Maurice úgy, hogy alhálózatokra osztja a hálózatot?*

V A hálózat szakaszolásának két előnye van. Először is, csökkenti az adatforgalmat. Másodszor, mivel az útválasztók nem fizikai, hanem logikai címeket használnak, az útválasztás módot ad arra, hogy egymástól eltérő fizikai felépítéssel rendelkező hálózati szakaszokat kössünk össze.

K *Miért használt Maurice DNS-kiszolgálót állomásleíró fájlok beállítása helyett?*

V Ha „hosts" fájlokat használt volna, Maurice-nak külön-külön kellett volna beállítania minden állomásleíró fájlt, ami nagyon sok időt vett volna igénybe. Ezenkívül az állomásleíró fájlokat minden alkalommal frissíteni kellett volna, ha valamilyen változás áll be a hálózatban.

TÁRGYMUTATÓ

A, Á

B

C, Cs

D

E, É

F

G, Gy

H

J

K

L

M

N, Ny

O, Ö

P

Q

R

S, Sz

T, Ty

U, Ü

V

W

X

Y

Z