**Fájlmegosztási szolgáltatások biztosítása**

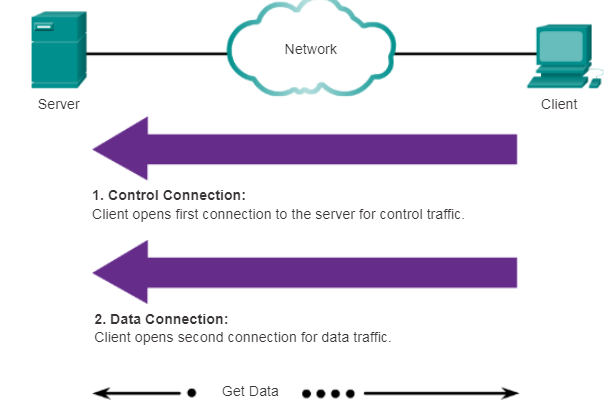
Egy másik általánosan használt alkalmazási rétegbeli protokoll a **fájlátviteli protokoll (File Transfer Protocol, FTP)**. **Az FTP-t egy kliens és egy szerver közötti adatátvitelre fejlesztették ki. Az FTP-kliens egy FTP-démont (FTPd) futtató szerverről való adatletöltésre vagy arra történő adatfeltöltésre szolgáló alkalmazás**.

Amint azt az ábra is mutatja, a kliens és a szerver közötti sikeres adatátvitelhez az FTP két kapcsolatot igényel. Egyet a parancsoknak és a válaszoknak, a másikat pedig a tényleges fájlátvitelhez.

Az adatátvitel mindkét irányba megtörténhet. A kliens tölthet le adatokat a szerverről és oda tölthet is fel adatokat.

* *Az első kapcsolatot a szerverrel a kliens kezdeményezi a kliens parancsaiból és a szerver válaszaiból álló vezérlési forgalomnak.*
* *Szintén a kliens kezdeményezi a tényleges adatátvitelre szolgáló második kapcsolatot is a szerverrel. Ez a kapcsolat minden egyes adatátvitel alkalmával létrejön.*

FTP folyamat



20-as port

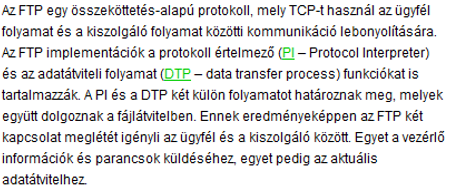
21-es port

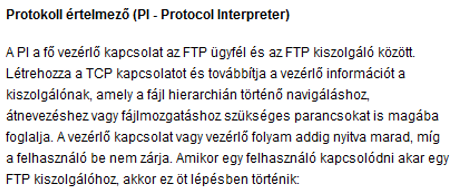
kliens nyitja meg az adatforgalom második kombinációját.

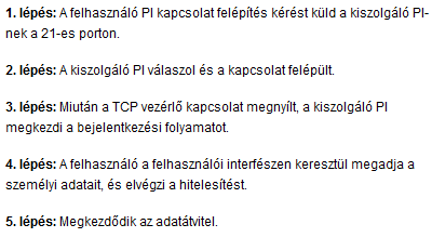
az ügyfél megnyitja az első kapcsolatot a szerverrel a forgalom ellenőrzése érdekében.

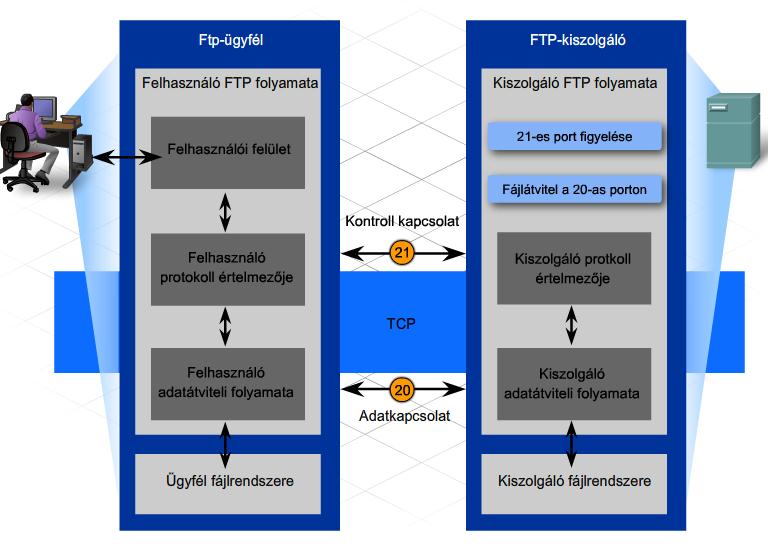
A vezérlő kapcsolaton keresztül küldött parancsok alapján az adatok letölthetők a szerverről

vagy az ügyfélről

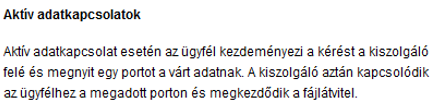


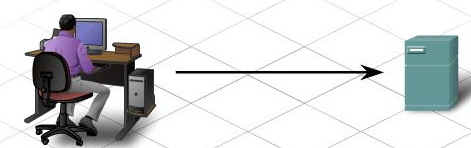


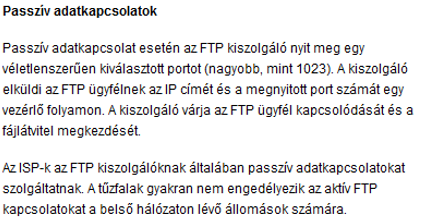


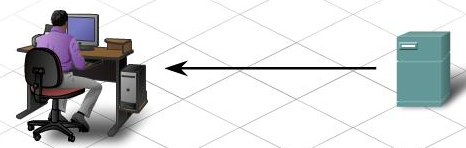












## Fájlmegosztási szolgáltatások biztosítása

**Az SMB (Server Message Block) egy az IBM által az 1980-as évek végén kifejlesztett kliens-szerver fájlmegosztó protokoll, amely olyan osztott hálózati erőforrások szerkezetét írja le, mint például könyvtárak, fájlok, nyomtatók és soros portok**. Ez egy kérés-válasz protokoll.

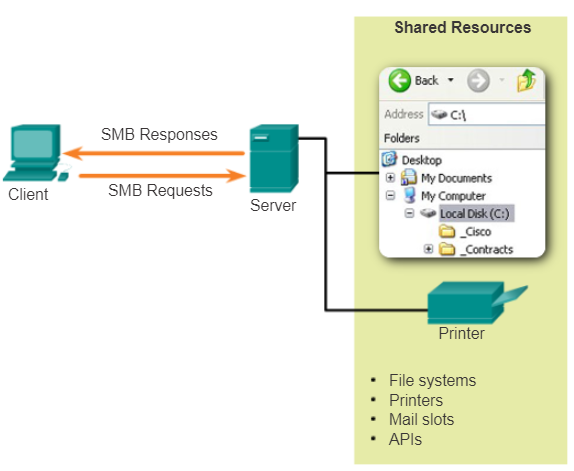
**Az SMB-protokoll leírja a fájlrendszer elérését, valamint azt, hogy a kliensek hogyan kérhetik le a fájlokat**. Leírja továbbá az SMB-protokoll folyamatok közötti belső kommunikációját is. **Minden SMB-üzenetnek közös a formátuma**. *Ez a formátum egy állandó hosszúságú fejlécet és egy azt követő változó méretű paraméter- és adatkomponenst használ.*

Az SMB-üzenetek az alábbiakra alkalmasak:

* **Párbeszédek indítása, hitelesítése és lezárása.**
* **Fájl- és nyomtatóelérés vezérlése.**
* **Egy alkalmazás és egy másik eszköz üzenetváltásainak biztosítása.**

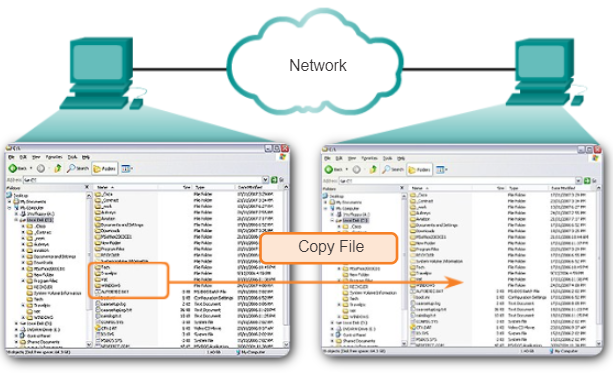
Az SMB alapú fájlmegosztás és nyomtatás a Microsoft legfőbb hálózati szolgáltatásaivá vált. A Windows 2000 szoftversorozat megjelenésével a Microsoft megváltoztatta az SMB által használt mögöttes struktúrát. **A Microsoft termékek korábbi verzióiban az SMB-szolgáltatások névfeloldásra még nem TCP/IP-re épülő protokollt használtak. A Windows 2000-től minden Microsoft termék már DNS neveket használ, ami közvetlenül támogatja az SMB-erőforrások TCP/IP protokoll alapú megosztását** (ábra).

**Szerverüzenet-blokk**



A 2. ábra két Windows PC közti SMB-alapú fájlcserefolyamatot szemlélteti.

**A kiszolgáló tömegesen blokkolja a fájlmegosztást**



**A fájlátviteli protokollal (FTP) ellentétben a fájlmegosztással a kliensek tartós kapcsolatokat építenek ki a szerverekkel.** A kapcsolat létrejötte után a kliens felhasználója a szerveren lévő erőforrásokat helyi erőforrásként tudja elérni.

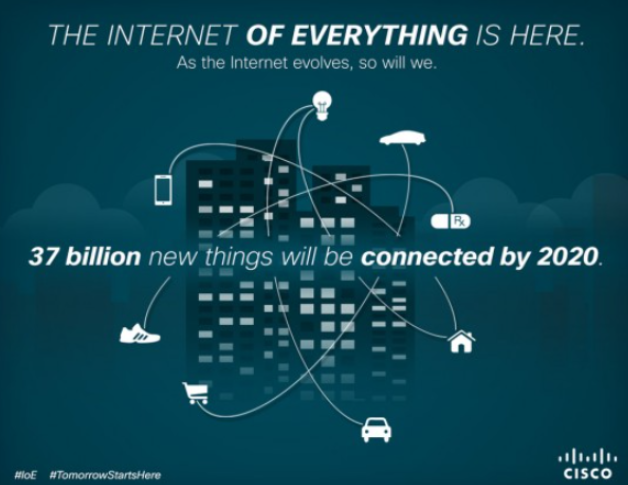
**A Linux és Unix operációs rendszerek egy SAMBA nevű SMB-verzió alkalmazásával biztosítanak módszert a Microsoft hálózatokkal történő erőforrás-megosztásához. Az Apple Macintosh operációs rendszerek ugyancsak támogatják az SMB-protokoll alapú erőforrás-megosztást.**

Az alkalmazási réteg felelős a hálózati kommunikációt kezelő és továbbító mögöttes folyamatok közvetlen eléréséért. Az adathálózat típusától függetlenül ez a réteg egyben a kommunikációk forrása és célja is. Valójában a hálózathasználat fejlődése egyben az alkalmazástípusok fejlesztésére is kihat.

**Az olyan trendek**, mint a „hozd a saját eszközöd” (Bring Your Own Device, BYOD), **a bárhonnan történő hozzáférés lehetősége, a virtualizáció, valamint a gép-gép közti közvetlen kommunikáció (Machine-to-Machine, M2M) újfajta alkalmazásoknak nyitottak utat**. *A becslések szerint 2020-ra mintegy 50 milliárd eszköz lett összekapcsolva. Egyedül 2010-ben több mint 350 000 alkalmazást fejlesztettek, több mint 3 millió letöltéssel.  Mindez az emberek ösztönös kapcsolatainak, folyamatoknak, adatoknak és tárgyaknak a világát eredményezi a hálózaton.*

**Az "okos" címkék használata, valamint a hétköznapi eszközök** – a kerékpároktól és palackoktól kezdve a hűtőszekrényeken át a járművekig – **digitálissá tétele és internetre csatlakoztatása új és szinte elképzelhetetlen távlatokat nyit** az emberek, de a vállalatok interakciójában is. *A tárgyak képesek lesznek információt gyűjteni, fogadni és küldeni felhasználóiknak és más eszközöknek i*s. Ahogy azt az ábra is szemlélteti, az internet fejlődésének **ez az új hulláma a tárgyak internete (Internet of Things, IoT)** vagy "minden a hálón" néven vált ismertté.

Mára több mint 100 millió árusító automata, jármű, füstérzékelő és egyéb eszköz oszt meg automatikusan információkat, és ez szám a [Berg Insight](http://www.berginsight.com/) piackutató számításai szerint 2016-ra eléri a 360 milliót*. Ma a fénymásolók egy M2M-modullal automatikusan képesek friss tonert és papírt rendelni, vagy egy hiba miatt riasztani a technikust, akár a javításhoz szükséges alkatrészt is meghatározva*.



37 milliárd új dolog fog összekapcsolódni 2020-ig.

Ahogy az internet fejlődik, úgy fogunk mi is.

Itt van mindennek az Internetje.

**Az alkalmazások számának robbanása** nagyrészt a hálózati adatfeldolgozás hatékony réteges szemléletének köszönhető. Különösen az alkalmazási réteg és az adattovábbítás funkcióinak elkülönítése segít abban, hogy **az alkalmazási protokollokat megváltoztathassuk, és új alkalmazásokat fejleszthessünk anélkül, hogy az adatok hálózaton keresztüli átjutásával külön foglalkozni kellene**. Ez a feladat más rétegekre, azaz más fejlesztőkre hárul.

*Animáció: 10.3.1.2*

Ahogy azt az ábra mutatja, **amikor egy alkalmazás egy kérést küld a szerveralkalmazásnak, akkor az alkalmazási réteg összeállítja az üzenetet, majd kézbesítésre továbbadja a kliens alsóbb rétegeinek. Ahogy az üzenet végighalad a protokoll-vermen, minden alsóbb réteg beágyazza azt, azaz ellátja a saját kommunikációs protokolljához tartozó fejléccel**. Ezeknek a küldő- és fogadóállomáson is meglévő protokolloknak az együttműködése teszi lehetővé az alkalmazások végponttól végpontig terjedő hálózati adattovábbítását.

Protokollok - mint például a HTTP - teszik lehetővé a weboldalak állomásokhoz való eljutását. Miután már tanultunk a különböző rétegekről és funkcióikról, végigkövethetjük egy weboldal webszerverről történő lekérését, hogy lássuk ezen független funkciók teljes együttműködését.

A TCP/IP modellt szerint a teljes kommunikációs folyamat hat lépésből áll:

* 1. **Az adatok létrehozása**

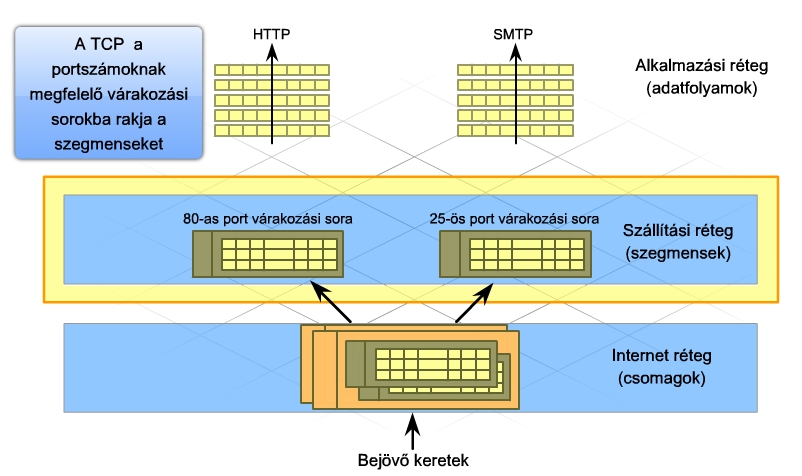
Az első lépés **az adatok létrehozása a kezdeményező forrásállomás alkalmazási rétegében**. Ebben az esetben a **webes kliens HTTP GET-kérésének összeállítása után az adatokat kódolják, tömörítik és szükség esetén titkosítják**. A TCP/IP modellben ez az alkalmazási rétegbeli protokoll feladata, mivel az **magába foglalja az OSI-modell alkalmazási-, megjelenítési- és viszonyrétegei funkcióit** is. Az alkalmazási réteg ezeket az adatokat adatfolyam formájában küldi tovább a szállítási rétegnek.

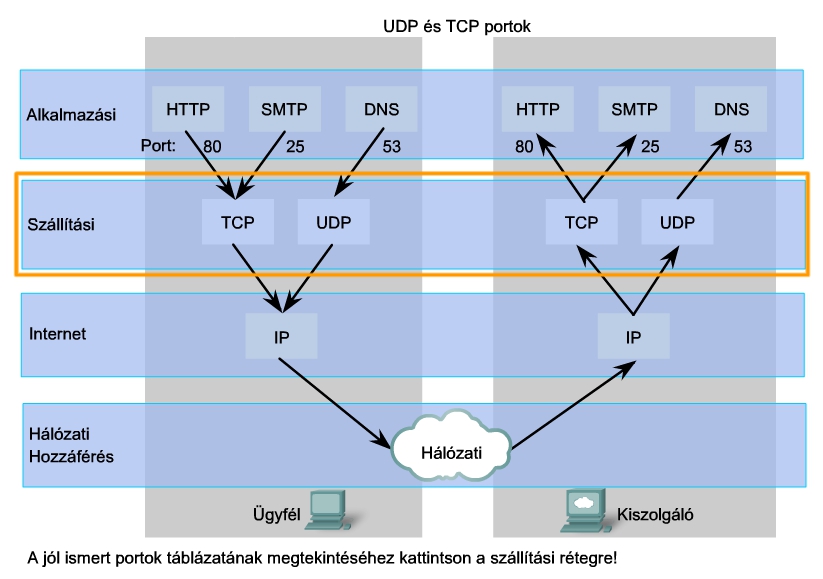
* 1. **Szegmentáció és kezdeti adatbeágyazás**

A protokoll-vermen lefelé haladva a **következő lépés az adatok szegmentációja és beágyazása. A szállítási rétegben a HTTP GET-üzenetet kisebb és kezelhetőbb darabokra tördelik, majd minden egyes darab egy szállítási rétegbeli fejlécet kap**. *A szállítási réteg fejlécében az üzenet újbóli összeállításához szükséges adatok vannak*. **Tartalmaz egy azonosítót is, a 80-as portszámot. Ez jelzi majd a célszervernek, hogy az üzenet a webszerver alkalmazásnak küldték. Hozzáadódik még egy véletlen generált forrásport azonosító is, hogy a kliens képes legyen a visszatérő kommunikációt a megfelelő kliensalkalmazásának továbbítani**.

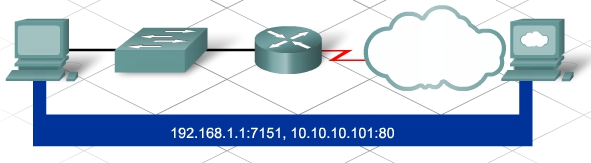
* 1. **Címzés**

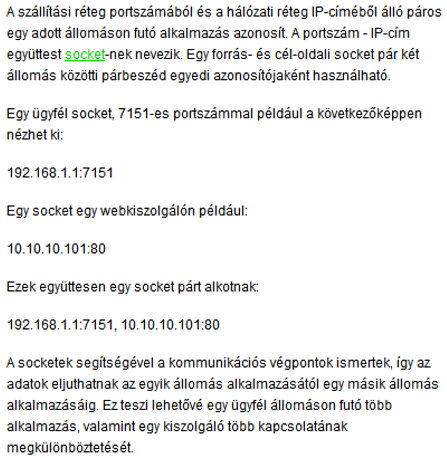
A következő lépésben, ahogy azt az ábra is mutatja, **a szegmensek címazonosítókat kapnak. Ahogyan az adatok célhoz történő továbbítását előkészítő protokolloknak is több rétege van, úgy a kézbesítést biztosító címzés is többrétegű**. **A hálózati réteg feladata az adatok forrástól a célállomásig történő eljutását biztosító címek hozzáadása**. A hálózati réteg ezt a szegmensek IP-csomagba történő beágyazásával oldja meg. **Az IP-csomag fejléce tartalmazza a forrás- és célállomások IP-címeit.** (A célállomás IP-címét általában egy tartománynév DNS lekérdezéséből nyerjük.) **A forrás és cél IP-cím, valamint a forrásport és a célport együttesét socket-nek nevezzük. A socket-et használjuk a kliens által kért kiszolgáló és szolgáltatás azonosítására.**





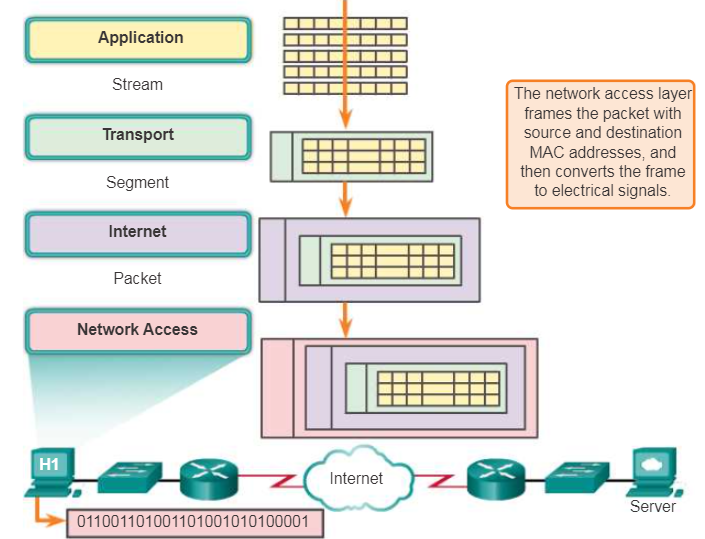
**Socket:**





* 1. **Az átvitel előkészítése**

**Miután a csomag megkapta az IP-címzést, azt a hálózatelérési réteghez továbbítják, hogy az adatok az átviteli közegre kerülhessenek** (lásd ábra). Azért hogy ez rendben megtörténhessen, a **hálózatelérési rétegnek a csomagot egy fejléccel és egy utótaggal ellátott keretbe kell ágyazni**a. **Ez a keret tartalmazza a forrásállomás, valamint a célállomás felé vezető útvonal következő állomásának (next hop) fizikai címét**. Ez megegyezik az OSI-modell második, azaz adatkapcsolati rétegének (Layer 2) funkciójával. A második réteg egy helyi hálózaton belüli végzi az üzenetek továbbítását. **A második rétegbeli cím az adott helyi hálózaton egyedi és a végberendezés címét reprezentálja a fizikai közegen**. **Ethernetet használó LAN-okban ezt a címet közeghozzáférés-vezérlési címnek** (MAC, Media Access Control) nevezik. A hálózatelérési réteg a forrás- és célcímekkel kiegészített keretet, bitekké, aztán pedig elektromos- vagy fényimpulzus jelekké alakítja, majd a fizikai közegen továbbítja.



* 1. **Az adatok átvitele**

**Az adatokat átviteli közegekből és közvetítő eszközökből álló összekapcsolt hálózatokon keresztül továbbítjuk**. Ahogy a beágyazott üzenet átjut a hálózaton, különböző közegeken és hálózattípusokon megy keresztül. **A hálózatelérési réteg határozza meg a különböző közegek keretformátumát és a keretek küldési módját, amit közeghozzáférés-vezérlésnek nevezünk**.

**Ha cél- és a forrásállomás ugyanazon a hálózaton van, akkor a csomag a két állomás között a helyi közegen forgalomirányító szükségessége nélkül kerül kézbesítésre. Ha azonban a cél- és a forrásállomás nem ugyanazon a hálózaton vannak, akkor a csomag továbbítása számos hálózaton, különböző közegtípusokon és forgalomirányítókon keresztül történhet**. **Ahogy a keret áthalad a hálózaton, a benne lévő információ nem változik**.

A helyi hálózatok határán egy közvetítő eszköz, általában egy forgalomirányító, kibontja a keretet, hogy a csomag fejlécéből kiolvashassa a célállomás címét. A forgalomirányítók ezen cím hálózatazonosító részét használják a célállomáshoz vezető út meghatározására. **Ahogy a forgalomirányító az útvonalat meghatározta, a csomagot új keretbe ágyazza és elküldi a célállomáshoz vezető út következő ugrásának címére.**

*Animáció: 10.3.1.5*

* 1. **Az adatok kézbesítése a megfelelő alkalmazáshoz**

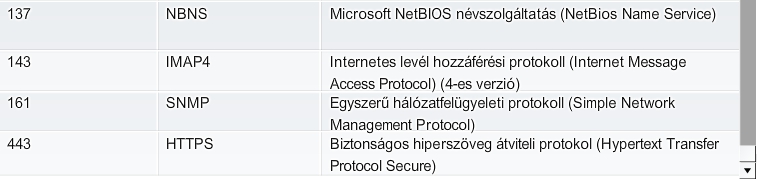
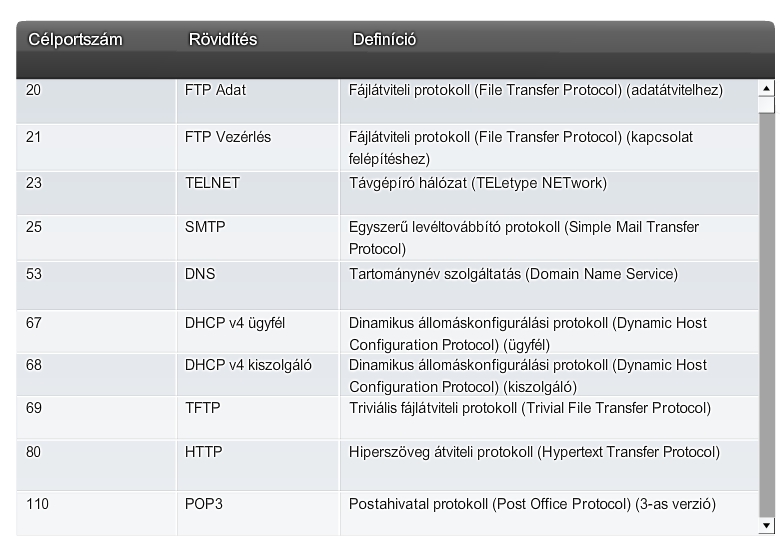
Végül a keret megérkezik a célállomáshoz. **Ahogyan az adat a célállomáson felfelé halad a protokoll-veremben, beágyazását visszafejtik, majd végül újból összeállítják**. A hálózatelérési rétegtől, a hálózati rétegen és a szállítási rétegen keresztül az adat folytonosan halad felfelé míg végül el nem éri az alkalmazási réteget, ahol feldolgozásra kerül. **De honnét tudja az eszköz, hogy az adatok a megfelelő alkalmazáshoz jutottak-e?**

Emlékezzünk rá, hogy a **szállítási rétegbeli PDU fejlécének adatai azonosítják a célállomáson azt a folyamatot vagy szolgáltatást, amelyhez az illető adat tartozik**. Az állomások, akár kliensek, akár szerverek az interneten, több hálózati szolgáltatást is képesek párhuzamosan futtatni. **Egy PC-n gyakran egyszerre van megnyitva egy levelezőkliens, egy böngésző, egy üzenetküldő program, valamilyen média folyam, esetleg egy játék. Ezek az önállóan futó programok lehetnek példák az egyedi folyamatokra.**

*Egy weboldal megtekintése legalább egy hálózati folyamatot igényel. Egy hiperlinkre kattintáskor a böngésző egy kommunikációs folyamatot kezdeményez egy webszerver. A háttérben ugyanakkor egy levelezőkliens e-maileket küldhet és fogadhat, egy kolléga, vagy barát pedig chat-elhet.*

Vegyünk egy olyan számítógépet, aminek csak egy hálózati interfésze van. A PC-n futó valamennyi alkalmazás adatfolyama ezen az egy interfészen keresztül lép be és távozik, ám a chat üzenetek mégsem jelennek meg az éppen szerkesztett dokumentum közepén és az e-mailek sem jelennek meg a játék felületén.

**Ez azért van így, mert a forrás- és célállomásokon lévő egyes folyamatok egymással kommunikálnak. Minden alkalmazást, vagy folyamatot egy 4. rétegbeli portszám jelöl**. Az egyedi párbeszédeket az egymással kommunikáló alkalmazások 4. rétegbeli forrás- és cél portszám párosa azonosítja**. Adatok beérkezésekor megvizsgálják a portszámot, hogy az adatok céljául szolgáló alkalmazást vagy folyamatot kiválaszthassák**.



**Portszámok:**

