

Hálózati ismeretek

Felels: Zsolt Tasnádi

CUR-56 - Hálózati ismeretek

UNDER REVIEW

- Az internet rövid története
 - Mainframe és Teletype
 - ARPAnet
 - BBS
 - Internet
- A számítógép-hálózat fogalma
 - A számítógép-hálózat céljai
 - Miért is érdemes számítógépeinket hálózatba kapcsolni?
 - Milyen veszélyeket rejt a gépek hálózatba kapcsolása?
- A szerver-kliens kapcsolat
 - Kliens
 - Kiszolgáló/server
 - A hálózat sebessége
 - Hoszt
 - Alhálózat
 - Hálózati hierarchia
 - Hoszt-terminál alapú hálózatok
 - Egyenrangú (peer to peer) hálózatok
 - Szerver-kliens hálózatok
- A hálózatok tipizálása
 - Lokális hálózat (Local Area Network, LAN)
 - Nagyvárosi hálózat (Metropolitan Area Network, MAN)
 - Nagy kiterjedésű hálózat (Wide Area Network, WAN)
- Topológia
 - Csillag topológia
 - 1. ábra: Csillag topológia
 - Gyűrű topológia
 - 2. ábra: Gyűrű topológia
 - Sín (busz) topológia
 - 3. ábra: Busz topológia
 - Fa topológia
 - 4. ábra: Fa topológia
 - Átviteli módszer alapján
 - Alapsávú vagy digitális átvitel
 - 6. ábra: Digitális átvitel
 - Szélessávú vagy analóg átvitel
 - 7. ábra: Analóg átvitel
 - Aszinkron kommunikáció
 - Szinkron mód
- A TCP/IP protokoll család
 - A TCP/IP protokollcsalád különféle protokolljai
 - A TCP/IP egyszersített rétegmodellje
 - TCP/IP Ethernet-csomag
 - IP-címek és útválasztás
 - IP-címek
 - Hálózati maszkok és útválasztás
 - Speciális címek
 - IP-címtartományok privát felhasználásra
- Alkalmazási réteg
 - Protokolljai
- A HTTP protokoll
 - Kérés (request)
 - Metódusok
 - Biztonságos metódusok
 - Idempotens metódusok
 - Feltételes kérés
 - Válasz (response)
 - Státuszkódok
 - Header sorok
 - Feltételes válasz
 - Hatékonyságnövelés
 - Munkamenet (session)
 - Biztonságos HTTP

- A HTTPS URI-séma
- A HTTP 1.1 Upgrade header
- Kapcsolódó anyagok

Az internet rövid története

Mainframe és Teletype

A nagyszámítógép, (angol kifejezéssel: mainframe, szakmai körökben elterjedt kifejezéssel nagygép vagy nagy vas) azokat a nagy és „drága”, nagy teljesítményű számítógépeket jelenti, amelyeket fleg kormányzati intézmények, nagyvállalatok, és bankok használnak az üzletileg kritikus alkalmazásaik futtatására. Általában nagy mennyiségű adat kezelésére, tárolására képesek, amelyeket a statisztika, a nyilvántartások, ERP rendszerek, és a pénzügyi tranzakció feldolgozási folyamatok igényelnek.

A kifejezés a korai 1970-es évekből ered, amikor kisebb, kevésbé komplex számítógépek kerültek piacra, (például a DEC PDP-8 és a PDP-11 sorozat) amelyek kisgépek, vagy *minik* néven lettek közismertek. Az ipari, banki, kormányzati felhasználók a fizikai valójukban is ténylegesen nagy gépeket, a „számítógépeket” kezdték nagyszámítógépként emlegetni.

ARPAnet

A számítógépek hálózattá való összeköttetésének gyökerei a hidegháború éveire nyúlnak vissza. 1957-ben a Szovjetunió fellötte Szeptnyik nevű műholdját. Válaszként 1958-ban az Egyesült Államok Védelmi Minisztériumán belül működő Korszerű Kutatási Programok Intézete létrehozta az ARPA (Advanced Research Projects Agency) nevű szervezetet, amelynek az volt a célja, hogy tudományos és technikai kutatásainak eredményeként az Egyesült Államokat vezető szerephez juttassa a hidegháborúban. Éppen ezért kísérletsorozatot folytattak, hogy létrehozzanak egy olyan számítógépes rendszert, amely ellenáll a különböző katasztrófáknak, az egész országot átfogja, és amelyben a katonaság üzeneteket, adatokat adhat és kaphat, bármi is történjék. Felmerült a kérdés, hogy egy esetleges nukleáris háború esetén az amerikai kormányzati szervek hogyan tudnák fenntartani a kommunikációt egymás között.

A RAND Corporation, az Egyesült Államok első számú hidegháborús szervezete a probléma megoldására egy olyan hálózatot képzelt el, amely képes városokat, államokat, bázisokat összekötni, de nincs központja, hiszen ha lenne, az lenne az első célpont, amit megsemmisítenének. A hálózatnak különböző pontokból kell állnia, a csomópontoknak egyenlő értékűnek kell lenniük, így önállóan alkothatnak, küldhetnek és fogadhatnak üzeneteket. Az üzenetek csomagokra oszlanak, minden csomagnak külön címe van. Minden csomag egy adott forráscsomóponttól indul, és egy adott célcsomóponthoz érkezik.

1967-ben megrendezték a RAND, az NPL (National Physical Laboratory) és az ARPA találkozóját, azzal a céllal, hogy továbbfejlesszék a hálózatot.

1968-ban az ARPA kezdeményezte, hogy az akkori szuperszámítógépek alkossanak egy olyan masszív hálózatot, amelyen keresztül hatékonyan lehet bonyolítani az országos kutatási és fejlesztési terveket. E masszív hálózat első csomópontját (node) a Kaliforniai Egyetemen állították fel 1969. szeptember elsején. A második csomópontot október elsején a Stanford Kutatóintézetben (SRI) kötötték be. Az év végére bekötötték a harmadik csomópontot a Kaliforniai Egyetem Santa Barbara-i (UCSB) egységében, és a negyediket az Utah-i Egyetemen. A négy csomópontból álló hálózatot ARPANET-nek keresztelték el.

Az ARPANET-nek köszönhetően a tudósok, nagy sebességű közvetítővonalakon, távolból is megoszthatták egymással eredményeiket. A csomópontok száma egyre nőtt: 1971-ben 15, 1972-ben már 37 működött. A hálózatra a NASA, valamint neves amerikai egyetemek is csatlakoztak.

1971-ben Ray Tomlinson kifejlesztette az e-mail programot, amely üzenetek küldését tette lehetővé a hálózat csomópontjai között. 1972. márciusban Tomlinson bevezette a @ jelet, mely az angol at prepozíciót jelölte (arra utalt, hogy az illetőt hol lehet elérni). Tomlinson írta a legelső e-mail üzenetet a világon. Egy irodában található két, az ARPANET-en keresztül összekötött számítógépen próbálta ki, hogy működik-e a rendszer. Az üzeneteket saját magának küldte. A legelső üzenet ezt tartalmazta: QWERTYUIOP. Ez nem más, mint a számítógép billentyűzetének a felső betűsora.

Tomlinson találmányának akkora sikere volt, hogy két évvel később az ARPANET-en bonyolított információcsere 75 százaléka e-mailen keresztül folyt már. Azonban még néhány évnek el kellett telnie ahhoz, hogy a fejlesztők felismerjék, hogy a számítógépes kommunikációra kifejlesztett hálózat közhasználatú, elektronikus postává válhat. Miután ezt felismerték, az emberek saját "account"-ot és személyre szóló e-mail címet szereztek az ARPANET-számítógépekhez, így megkezdődhetett a privát levelezés, csevegés a hálózaton keresztül. Megjelentek a hírlevelek is, amelyeket egyszerűen több felhasználónak tudtak elküldeni.

1979-ben már felbukkantak az úgynevezett "emoticon"-ok (más néven smiley) Kevin MacKenzie javaslatára (túl unalmasnak találta az e-mailek száraz, egyszerű felületét), melyek akár e-mailben, akár csevegéskor az érzelmek egyszerű kifejezésére alkalmasak.

Az ARPANET másik hatalmas előnye az volt, hogy bármilyen számítógépet hozzá lehetett kapcsolni az új csomagkapcsolt hálózathoz, amely ismerte annak közvetítőnyelvét. Az ARPA kezdeti kommunikációs szabványrendszere az NPC (Network Control Protocol) volt, melyet felváltott a TCP/IP (Transmission Control Protocol), mely az üzeneteket keletkezési helyükön csomagfolyamokká alakítja, majd újra üzenetté rendezi őket a célállomáson. Az IP (Internet Protocol) a címzésért felel, vagyis azért, hogy a csomagok többféle csomóponton és többféle hálózaton haladjanak keresztül, amelyek a legkülönbözőbb szabványrendszerek alapján működnek.

1981-ben létrejött a CSNET (Computer Science Network) azzal a céllal, hogy olyan egyetemi tudósok is hozzáférjenek a hálózaton található

információkhoz, akiknek nincs ARPANET-elérhetőségük. A CSNET később a Számítógép és Tudomány Hálózataként lett ismert.

1983-ban az ARPANET-ről levált a MILNET elnevezésű katonai szegmens. Az akkor létező 113 csomópontból 68 a MILNET hatásköre alá került.

1984-ben a Nemzeti Tudományos Alap (National Science Foundation, NSF) tovább akarta fejleszteni a hálózatot, magasabb szintű technikát alkalmazva. Az NSFNET nagyobb tudású számítógépeket csatlakoztatott egymáshoz, növelte a telefonvonalak teljesítményét. A hálózatok egyre szaporodtak és ezzel megkezdődött az Internet terjesztése az egyetemek felé. Ezzel párhuzamosan a NASA és más állami ügynökségek is "bekapcsolódtak" az Internet világába. A külföldi számítógépeket elhelyezkedésük alapján jelölték meg, a többi számítógépet a hat fő internetágazat szerint csoportosították: gov, mil, edu, com, org és net (1985).

Az ARPANET 1989-ben formálisan megszűnt, de az a folyamat, amit elindított, tovább fejlődött.

BBS

A Bulletin Board System (BBS) egy olyan informatikai rendszer, ami egy adott kör – gyakran bárki – számára elérhetlen biztosítja az üzenetek váltását illetve az adatcserét. A BBS rendszerek az 1980-as évektől kezdve terjedtek el, népszerűségük az internet elterjedése miatt a 90-es évek végére nagyban lecsökkent, noha még napjainkban is számtalan működő rendszert találunk, részben az internet által biztosított olcsó adatkapcsolatokat igénybe véve.

A BBS rendszereket terminálprogramok segítségével lehet – azokra csatlakozva – használni. A kapcsolat módja változatos lehet éppúgy, mint a használt terminálprogram fajtái.

A BBS-ek fénykorukat a telefonos modemek elterjedésének köszönheték. A 80-as évek közepétől kezdve számtalan telefonon át felhívható rendszerbe lehetett távoli számítógépekről bejelentkezni, és ott elektronikus leveleket, nyilvános üzeneteket váltani, illetve adatokat (programok, képek, és így tovább) cserélni. A modemek sebességének növekedésével (és a telefonvonalak minőségi jellemzőinek javulásával, illetve a telefon elfizetések számának növekedésével) egy-egy helyi BBS rendszert több száz, akár több ezer felhasználó is látogatott.

Az USA-beli elterjedésüket segítette az is, hogy ott a helyi telefonhívás teljesen ingyenes, így a BBS-eket gyakorlatilag költségmentesen lehetett használni. Ez hozta létre az igényt, hogy országos lefedettségű hálózatokká szervezve a BBS rendszerek egymással képesek legyenek kommunikálni, és mindig a legolcsóbb módon, a közeli rendszereken át továbbítsák az üzeneteket, amit a helyi felhasználók ingyen el tudnak érni; így alakult meg a FidoNet, majd később számos más, hasonló hálózat.

A BBS-ek nem csak telefonvonalon, de egyéb adatkapcsolatokon, mint például az amatr rádiós hálózat, vagy később a telenet kapcsolat is elérhetők voltak.

Az internet elterjedése olcsó adatkapcsolatot hozott a világban, így a BBS-ek telefonos elérhetősége háttérbe szorult, valamint a csak terminálon elérhető szolgáltatások kényelmetlenebbnek bizonyultak, mint az internet által biztosított számtalan elérési módszer (mint amilyen a World Wide Web vagy az FTP).

Mosaic

A **Mosaic** az NCSA (*National Center for Supercomputing Applications*) által fejlesztett grafikus webböngésző. A fejlesztése 1992-ben kezdődött, és 1997. január 7-én ért véget.

A Mosaic a 90-es évek egyik meghatározó internetes alkalmazása volt, mivel ez volt az első olyan fejlesztés, ami az internet osztott információk szolgáltatásait multimédiás grafikus felhasználói felületen jelenítette meg. Gyors elterjedése jelentősen hozzájárult a World Wide Web népszerűségéhez, amely eltt az internet többnyire csak FTP, Usenet és Gopher protollokon működött, főképp akadémiákon és kutatási intézményekben.

Internet

A legnagyobb előrehaladást a napjainkra járványként terjedő világhálózat, az Internet hozta. A fejlődés következtében a táv-adatátviteltől, terminál-számítógép kapcsolattól eljutottunk a videokonferenciáig, s ezzel még biztosan nem ért véget a hálózatok fejlődésének a korszaka. Az Internet a számítógépek világhálózata, amely milliók számára teszi lehetővé az adatok azonnali cseréjét. Életünket szinte már elképzelhetetlennek tartjuk nélküle, hiszen óriási mértékben megkönnyíti az információszerezést.

Az Internet 1983-ban kezdte működését, amikor az ARPANET két elkülönült hálózatra bomlott: MILNET és ARPANET. Mindkettő kapott hálózati címet (network number), és átjárókat (gateway) tettek közéjük, amelyek információcsomagokat tudnak továbbítani egyik-ről a másikra.

Ekkor a DCA (Defense Communications Agency) meghatározta, hogy minden ARPANET hoszt a TCP/IP protokollt használja. Mivel így az információ-csomagok formája szabványosítva lett, ez azt jelentette, hogy új hálózatokat lehet hozzákapcsolni a rendszerhez anélkül, hogy az eddigi felépítést meg kellene változtatni. Az Internet tehát az ARPANET-ből indult, s magába foglalt más hálózatokat, úgymint az NSFNET-et, a NEARNET-et s másokat. Több más hálózat, mint például a BITNET kapcsolódik az Internethez, de nem része annak.

Később az NSFNET egyetemek és kereskedelmi intézmények szuperszámítógépei között teremtett nagykapacitású összeköttetést, amelyek köré helyi hálózatokat szerveztek. Így az NSFNET is a "hálózatok hálózata" lett, mint az Internet, amelynek részét képezi.

Ahogy az évek telnek, úgy kezd a tudományos jelenlét az Interneten egyre inkább háttérbe szorulni azzal szemben, hogy a felhasználók özönlépi el a hálót, akik információkhoz, kommunikációhoz, szórakozáshoz vagy üzleti lehetőségekhez akarnak jutni az Internet révén. Az Internet boven meghaladta eredeti céljait, amelyeket a 60-as években elképzelték. Mára egy nagy világhálózattá nőtt ki magát, amelyhez a csatlakozó számítógépek száma túllépi az egymilliót. Ezen a legváltozatosabb intézmények számítógépeit kell érteni, pl. egyetemek, kutatóintézetek, számítógépgyártó cégek, kormányzati-államigazgatási hivatalok, rendőrségek, könyvtárak stb.

A számítógép-hálózat fogalma

A számítástechnika rohamos fejlődése a számítógépek széleskörű elterjedésével, s ezáltal a felhasználók táborának növekedésével járt együtt. A számítógépek összekötése iránti igény először akkor merült fel, amikor egyes csoportok némely erőforrást, azaz háttértárolót, nyomtatót, adatbázist vagy programot közösen szerettek volna használni. Ehhez szükség volt a számítógépek fizikai összekapcsolására, valamint néhány olyan gépre, amely rendelkezett ezekkel az erőforrásokkal, és így ezeket a csoport minden tagja ugyanolyan formában tudta használni.

A számítógép-hálózatok kialakítását követelő másik kihívás a helyi hálózatok összekapcsolása annak érdekében, hogy lehetővé váljon üzenetek, elektronikus levelek, valamint nagy adattömegek gyors és megbízható továbbítása akár kontinensek között is. Ugyancsak célszerűnek látszott lehetővé tenni, hogy egy-egy szuperszámítógép kapacitását ne csak a rákapcsolt gépekről lehessen használni, hanem megfelelő jogosultság esetén a világ távoli pontjairól is hozzá lehessen férni. Ezek után tisztázzuk a számítógépes hálózat fogalmát:

A hálózatok önállóan is működőképes számítógépek elektronikus összekapcsolása, ahol az egyes gépek képesek kommunikációra külső beavatkozással nélkül.

A számítógép hálózat olyan függőségben lévő vagy független számítógépek egymással összekapcsolt együttese, amelyek abból a célból kommunikálnak egymással, hogy bizonyos erőforrásokon osztozkodhassanak, egymásnak üzeneteket küldhessenek, illetve terhelésmegosztást vagy megbízhatóság növekedést érjenek el.

A számítógép-hálózat céljai

Miért is érdemes számítógépeinket hálózatba kapcsolni?

- Közösen használhatóak a hálózatban lev erőforrások – programok, adatok, perifériák – a felhasználók számára fizikai elhelyezkedésüktől függetlenül elérhetek, korlátot csupán a felhasználók jogosultsága jelenthet.
- Egyenletesebb teljesítmény-megosztással üzemeltethetők a rendszer erőforrásai.
- A rendszer nagyobb megbízhatósággal üzemeltethető. A hálózatba kapcsolt valamelyik periféria meghibásodása, leállása nem kell, hogy törvényszerűen az egész rendszer üzemképtelenségét okozza, mivel másik periféria átveheti a szerepét. A megbízhatóság nemcsak a hardver elemekre de a programokra, adatokra is kiterjeszthető, mivel redundáns módon több gép háttértárolóján is tárolhatók.
- A rendszer teljesítményének skálázható növelése. Ez azt jelenti, hogy rendszerünk teljesítménye fokozatosan, a jelentkező igények szerint növelhető, úgy hogy újabb erőforrásokat, processzorokat adunk hozzá.
- Távoli adatok elérése. A hálózat segítségével távoli adatbázisok elérése is lehetővé válik, úgy is hogy mi bővítsük az adatbázist, vagy csak lekérdezzük belőle. (PI; banki tranzakciók, vagy adatok lekérdezése, on-line katalógusok megtekintése, elektronikus vásárlás, stb)
- A számítógép-hálózat kommunikációs közegként is használható. A konvergencia jegyében tetten érhető, hogyan változott a hírközlésünk a hálózatba kapcsolt számítógép elterjedésével. A hirtelen megnövekedett digitális átvitelre használjuk, telefonrendszerünket jelentős részben digitális átvitelre használjuk.
- A személyek közötti kommunikációra is lehetőséget ad a hálózat. Példa erre az elektronikus levelezés, az IP alapú telefonálás, vagy az Internetes csevegés is.
- Az interaktív szórakoztatás alapját szintén a tartalmak digitalizálása és egységes tárolása, továbbítása teremtette meg. A továbbított adatok tartozhatnak hang-, vagy képi -, esetleg videó információhoz, a hálózat képes azokat átvinni, amennyiben elég nagy a kapacitása. Egyre több cég gyárt kimondottan otthoni szórakoztatásra szánt hálózati eszközöket, pl. otthoni médiaszervert az MP3-as hanganyagaink, vagy különböző formátumú videóink tárolására, és úgynevezett show-center eszközöket azok TV készülékeken való megjelenítésére.

Milyen veszélyeket rejt a gépek hálózatba kapcsolása?

- A hálózat kritikus pontjainak sérülése a rendszer használhatatlanságát eredményezheti.
- Kell védelem nélkül a rendszerből illetéktelenek is információhoz juthatnak.
- A koncentrált program, de különösen az adattárolás a felhasználók bizonyos kiszolgáltatottságát eredményezheti.
- A hálózatok nem-kívánt kellemetlen mellékhatása a számítógépes bűnözés terjedése: adatlopások, számítógépvírus fertőzések, rendszerbénítások.

A szerver-kliens kapcsolat

A kliens-szerver (magyarul: ügyfél-kiszolgáló) kifejezést először az 1980-as években használták olyan számítógépekre (PC-kre) amelyek hálózatban működtek. A ma ismert modell a 80-as évek végén vált elfogadottá. A kliens-szerver szoftverarchitektúra egy sokoldalú, üzenetalapú és moduláris infrastruktúra amely azért alakult ki, hogy a használhatóságot, rugalmasságot, együttműködési lehetőségeket és bővíthetőséget megnövelje

a centralizált, nagyszámítógépes, idosztásos rendszerekhez képest.

Kliens

A kliens (angolul client) olyan számítógép amely hozzáfér egy (távoli) szolgáltatáshoz, amelyet egy számítógép hálózathoz tartozó másik gép nyújt. A kifejezést először önálló programmal nem rendelkező végkészülékekre illetve terminálokra alkalmazták amelyek legfontosabb szerepe az volt, hogy a hálózaton keresztül kapcsolatba lépjenek az idosztással működő nagyszámítógépekkel és elérhetővé tegyék azok szolgáltatásait.

Jellemzi

- Kéréseket, lekérdezéseket küld a szervernek
- A választ a szervertől fogadja.
- Egyszerre általában csak kis számú szerverhez kapcsolódik
- Közvetlenül kommunikál a felhasználóval, általában egy GUI-n (Graphical User Interface = Grafikus felhasználói felület) keresztül

Kiszolgáló/server

A kiszolgáló vagy szerver (angolul server) olyan (általában nagy teljesítményű) számítógépet, illetve szoftvert jelent, ami más gépek számára a rajta tárolt vagy előállított adatok felhasználását, a kiszolgáló hardver erőforrásainak (például nyomtató, háttértárolók, processzor) kihasználását, illetve más szolgáltatások elérését teszi lehetővé

Jellemzi

- Passzív, a kliensektől várja a kéréseket
- A kéréseket, lekérdezéseket feldolgozza, majd visszaküldi a választ
- Általában nagyszámú klienshez kapcsolódik egyszerre
- Általában nem áll közvetlen kapcsolatban a felhasználóval

A kiszolgálókat többféleképpen csoportosíthatjuk, például:

- a funkciójuk szerint, például webkiszolgálók, FTP-kiszolgálók, adatbázis-kiszolgálók;
- a kiszolgált kör alapján, például internetes kiszolgálók, intranetes kiszolgálók;
- a teljesítményük alapján.

A hálózat sebessége

A különböző hálózatok adatátviteli sebessége eltérő, sőt nagyságrendbeli eltérés is elképzelhető, amit az adott hálózati struktúra, topológia határoz meg. Nagy szerepet játszik a számítógép-hálózatok átviteli képességének az alakulásában a hálózat topológiája, az adatátvitelt vezérlő protokoll típusa, a hálózat strukturális kialakítása, s nem utolsósorban az aktív, passzív eszközök sebessége is. E paraméterek határozzák meg a hálózat maximális átviteli sebességét.

A számítógép-hálózatok sebességének a mértékegysége **bit/sec**, illetve **bps**. Az egy másodperc alatt átvitt bitek számát határozza meg. Ez egy elméleti érték, mellyel a hálózat maximális átviteli képessége határozható meg.

A számítógépben az információ alapegységei a bit, digit, byte. Egy bit (**binary digit**) 0 vagy 1 értékű lehet, melynek 1 - 1 feszültségi szint felel meg. A digit 4 bitből áll, a byte 8-ból. A tárolókapacitást általában byte-okban szoktuk megadni. Magasabb mértékegységei:

1	K b y t e	(k i l o - b y t e)	= 1 0 2 4	b y t e
1	M b y t e	(m e g a - b y t e)	= 1 0 2 4	K b y t e
1	G b y t e	(g i g a - b y t e)	= 1 0 2 4	M b y t e .
1	T b y t e	(t e r a - b y t e)	= 1 0 2 4	G b y t e

Az átviteli sebességtől meg kell különböztetni a baud-ban mért sebességet, amely a másodpercenkénti jelváltások számát adja meg. 1 baud = 1 jelváltás/s. A jelsorozat sebessége és az adatátviteli sebesség számértéke csak akkor egyezik meg, ha az alkalmazott jelkódolási eljárás egy-egy bitnek egy-egy jelváltást feleltet meg.

Hoszt

Számítógépes hálózaton lévő "munkaállomás", amely valamilyen szolgáltatást is nyújt egy hálózatba építve. Itt futnak a felhasználói programok, helyezkednek el az adatbázisok. Ezeket a gépeket kommunikációs alhálózatok kötik össze, amelyek feladata a hosztok közötti kommunikáció megvalósítása, azaz az üzenetek továbbítása.

Alhálózat

A hálózat az összekötött alhálózatok összessége. Az alhálózatokat kapcsolóeszközök (routerek) kötik össze egymással. Az alhálózat adattovábbítás szerint lehet pont-pont kapcsolat vagy broadcast típusú.

Két pont közötti csatornával rendelkező alhálózat (pont-pont kapcsolat)

A két kommunikációs végpontot egy kábellel kötik össze, és az üzenetek ezen a kábelén keresztül haladnak. Amikor egy vevő megkapja a csomagot, és az nem neki szól, akkor továbbadja egy következő pont-pont összeköttetésen keresztül. Pont-pont összeköttetésekből felépülő nagyobb hálózat struktúrája lehet: csillag, gyűrű, fa, teljes vagy szabálytalan. A gyűrű topológia a működési biztonságot növeli. A fastruktúra előnye a hosztok (alhálózatok) közötti minimális routerszám, hátránya a forgalom torlódása. A teljes összeköttetésű hálózat mindkét elonnyel rendelkezik, azonban rendkívül drága.

A pont-pont kapcsolatokból felépülő hálózatokkal kapcsolatos további fogalmak: Tárol és továbbít (store-and-forward)

A csomag a routerben tárolásra kerül, amíg a következő adatvonalon lehetőségessé nem válik a továbbítás.

Packet switch

Az adatcsomag tartalmazza a célállomás címét, ennek alapján a router (csak) a megfelelő adatvonalon továbbítja a csomagot.

Broadcast típusú alhálózat

Más néven üzenetszórásos alhálózatnak is nevezik. Jellegzetessége, hogy egy csatornán (átviteli közegen) sok állomás osztozik. Minden adás minden eszközhöz eljut. Az állomások a csomagban lévő cím alapján választják ki a nekik szólókat, a többit figyelmen kívül hagyják. Topológiai kialakítása lehet busz, gyűrű vagy csillag.

Hálózati hierarchia

Aszerint, hogy hálózatunkban milyen viszonyban állnak egymással a számítógépek, három típusra oszthatjuk őket.

Hoszt-terminál alapú hálózatok

A hálózat magját egy vagy több, egymással összeköttetésben lévő központi számítógép (hoszt) alkotja. Itt fut az operációs rendszer. Ehhez kapcsolódnak hozzá az intelligencia nélküli (buta) terminálok, amelyek egy billentyűzetből és egy képernyőből állnak. Ezen a hálózattípuson futnak a legbonyolultabb és legrégebben fejlesztett rendszerek.

Egyenrangú (peer to peer) hálózatok

A hálózatba kötött számítógépek közül bármelyik lehet kiszolgálója a többinek, amelyek a felajánlott erőforrást beépíthetik a saját rendszerükbe. Általában LAN-ok kialakításánál alkalmazzák, ahol viszonylag kevés a gép, a hálózati forgalom kicsi. Előnyei az olcsóság, egyszerűség. Hátránya a kis kapacitás, nagy feladatok elvégzéséhez nem vagy korlátozottan használhatók.

Szerver-kliens hálózatok

Ötvözik a peer to peer hálózatok olcsóságát, egyszerűségét a hoszt-terminál hálózatok nagy teljesítőképességével. Ebben a hálózatban találunk kiemelt számítógép-et (szerver), amely csak a kérések kiszolgálásával van elfoglalva. Itt fut a hálózati operációs rendszer. Az alkalmazói programok futtatása a kliens gép-ek feladata. A felhasználó által megfogalmazott kérések az alkalmazói programon keresztül eljutnak a szerver operációs rendszeréhez, amely ezen kéréseket kiszolgálja. Előnye: nem kíván nagyon komoly hardvert, gyors a kiszolgálás sebessége. Üzemeltetése olcsó. Nagy a szoftver ellátottság. Hátránya: az alkalmazói program a kliens gépen fut, így nagy a hálózati forgalom.

A hálózatok tipizálása

Topográfián a munkaállomások területi elhelyezkedését és összeköttetését értjük, tehát azt, hogy fizikailag hol találhatóak a hálózat végpontjai. Földrajzi elhelyezkedés szerint 3 féle hálózatot különböztetünk meg:

Lokális hálózat (Local Area Network, LAN)

Kis kiterjedésű, egyszerű szervezéssel meghatározott távolságon belül (maximum 10 km), azaz egyetlen épületen belül teszi lehetővé az információ és az erőforrások megosztását a felhasználók számára. A lokális hálózatban az eszközök a hálózat fizikai kialakítására telepített kábelben, vagy más átvivő közegen keresztül közvetlenül kapcsolódnak egymáshoz. Ez ellentétes a távolsági hálózatokkal, amelyek gyakran a nyilvános távközlés-technikai berendezéseket, a vonalkapcsolt vagy csomagkapcsolt adathálózatokat veszik igénybe kommunikációs csatornaként. Ebből következik, hogy keveset hibáznak és kicsi a késleltetésük. Sebességük 10 Mb/s (megabit / másodperc) és 100 Mb/s között mozog, de ma már előfordul az optikai kábeleknek köszönhetően, hogy a 100 Mb/s-os adatátviteli sebességet is elérhetik

Nagyvárosi hálózat (Metropolitan Area Network, MAN)

Nagyobb távolságra lévő gépek, LAN hálózatok összeköttetéséből alakul ki. Felépítése a LAN-okhoz hasonlít. Összeköt egymáshoz közel fekvő vállalati irodákat vagy akár egy egész várost. Egyik tipikus alkalmazása a világhálózat kiinduló pontjaihoz való belépésének biztosítása. Hatótávolsága 1 és 50 km között van.

Nagy kiterjedésű hálózat (Wide Area Network, WAN)

Egymástól nagy távolságra elhelyezkedő hálózatokat köt össze, akár az egész világot behálózhatja. A helyi hálózatok több millió bit/s-os átviteli sebességéhez képest a nagy távolságokra szolgáló átviteli közeg, és az átviteli sebesség sokkal kisebb.

Topológia

A számítógépek kábelezését néhány jellegzetes mértani formával szokás jellemezni, mint csillag, sín, gyűrű, fa vagy szabálytalan alak. Ennek megfelelően beszélhetünk csillag, sín, gyűrű, fa topológiákról. Ha a felsorolt elrendezési módú hálózatok közös hálózati kialakításban szerepelnek, hibrid hálózatról beszélhetünk. Topológián tehát a hálózat alkotórészeinek összekapcsolási módját, fizikai elrendezését, a hálózati eszközök összeköttetésének rendszerét értjük.

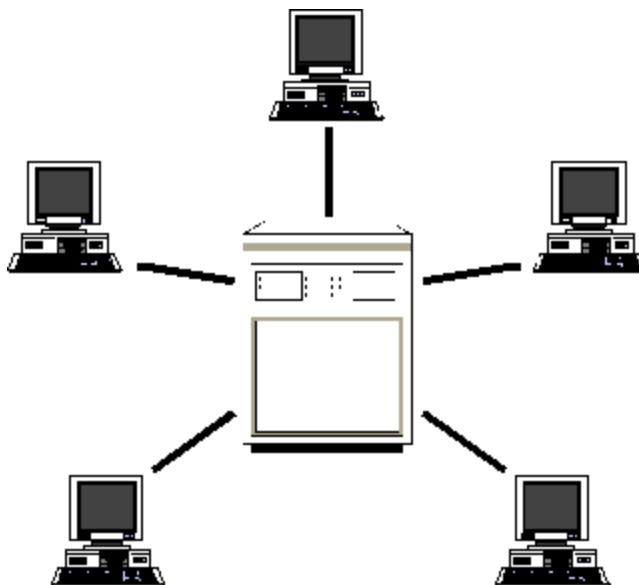
Csillag topológia

A legelső topológiák közé tartozik, mivel ezáltal könnyen megoldható volt a korai időkben az általánosan elterjedt központosított vezérlés. A csillag topológia esetén a munkaállomások közvetlenül tartanak kapcsolatot a szerverrel, így a központi erőforrások gyorsan és egyszerűen elérhetők. Ha nincs szükség folyamatos adatátvitelre, akkor a csomagkapcsolt eljárást alkalmazzák, különben pedig a klasszikus vonalkapcsolást. Ha az egyik számítógép kapcsolatba akar lépni a hálózat egy másik számítógépével, akkor a központi vezérlő létrehozza az összeköttetést, vagy legalábbis kijelöli a másik berendezés elérési útvonalát, s miután ez megtörtént, elkezdődhet a kommunikáció. Az összeköttetést követően az információcsere úgy bonyolódik le, mintha közvetlen kapcsolatban állna egymással a két számítógép. Ekkor a központi vezérlőnek már nincs feladata, tehát mintegy közvetítőként működik.

Ezt a központi vezérlő berendezést nevezik **HUB**-nak. A szakirodalom a hálózat egyes számítógépeit csatlakozási pontnak, angol kifejezéssel **node**-nak nevezi. A csillag topológia esetén az adatcsomagok az egyes csatlakozási pontoktól a központi hub felé haladnak. A központi hub az adatcsomagokat rendeltetési helyük felé továbbítja. Egy hub-ot használó rendszerben nincs közvetlen összeköttetés a számítógépek között, hanem az összes számítógép a hub-on keresztül kapcsolódik egymáshoz. Minden node egyetlen kábelben csatlakozik a hub-hoz. Mivel mindegyik gép külön kábelben csatlakozik a hub-hoz, ezért meglehetősen sok hálózati kábelre van szükség, ami adott esetben drágává teheti a telepítést. A csillag topológiához használhatunk árnyékolatlan **csavart érpáru huzalt (UTP)** vagy **árnyékolt csavart érpáru huzalt (STP)**. (ld. Vezetékes átviteli k ö z e g e k)

A csillag elrendezés egy összetettebb változata a hópehely (snowflake) topológia, amely nagyobb kiterjedésű hálózatok esetén több csillag topográfiájú hálózat kapcsolatát biztosítja úgy, hogy a hálózatok közé egy közös csomópontot, egy újabb központi vezérlőt iktat, ami lehetővé teszi két különböző hálózatban levő gép összeköttetését.

A csillag topológia legfőbb előnye az, hogy ha megszakad a kapcsolat a hub és bármelyik számítógép között, az nem befolyásolja a hálózat többi csomópontját, mert minden node-nak megvan a saját összeköttetése a hub-bal. A topológia hátránya az, hogy a központ meghibásodásával az egész hálózat működésképtelenné válik. Másik hátránya, hogy ha az egyik gép üzen a másiknak, előbb a központi gép kapja meg a csomagot, majd azt a célállomásnak továbbítja. Emiatt a központi gép gyakran túlterhelt. Strukturált kábelezéssel csökkenthető a központi gép és a hálózati szegmensek leterheltsége.



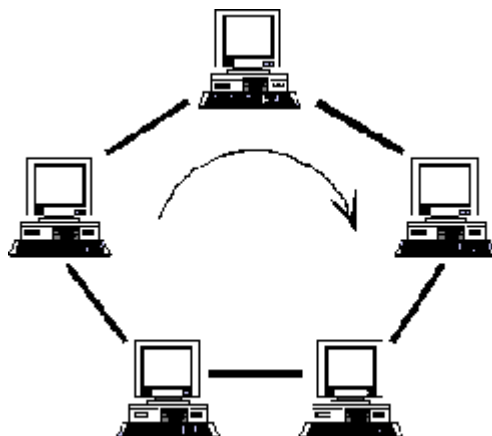
1. ábra: Csillag topológia

Gyűrű topológia

Minden állomás, beleértve a szervert is, két szomszédos állomással áll közvetlen kapcsolatban. Az összeköttetés körkörös, folyamatos gyűrű (megszakítás nélküli, de szükségszerűen kört képező), ebből következően a hálózatnak nincs végcsatlakozása. Bármely ponttól elindulva végül visszatérünk a kiindulóponthoz, hiszen az adat csak egy irányban halad. Az üzeneteket a gépek mindig a szomszédjuknak adják át, s ha az nem a szomszédnak szól, akkor az is továbbítja. Addig vándorol az üzenet gépről gépre, amíg el nem érkezik a címzethez. Mindegyik csomópont veszi az adatjelet, elemzi az adatokat, és ha az üzenet másik gép részére szól, akkor az adatokat a gyűrű mentén a következő géphez továbbítja. Az adatfeldolgozás cím alapján történik, azaz csak a címzett dolgozza fel az adatot, a többiek csak továbbítják.

A csillag topológiától eltérően a gyűrű topológia folyamatos útvonalat igényel a hálózat összes számítógépe között. A gyűrű bármely részén fellépő meghibásodás hatására a teljes adatátvitel leáll. A hálózattervezők a meghibásodások ellen néha tartalék útvonalak kialakításával védekeznek. Ezenkívül hátránya még az is, hogy az adat a hálózat minden számítógépén keresztülhalad, és a felhasználók illetéktelenül is hozzájuthatnak az adatokhoz.

A gyűrű alakú topológia esetén a hálózati kommunikáció lehet csomagkapcsolt és vezérelve alapján működő. Ezen az elven működik a [vezérelt gyűrű \(Token Ring\)](#), amit egy későbbi fejezetben fogok részletesen ismertetni. Mégis hogy ne lebegjen üresen a levegőben ez a fogalom, egy-két szóval ismertetem a lényegét. Itt egy vezérelt kering körbe a vonalon, és csak az a gép küldhet üzenetet, amelynél éppen a vezérlő van. A küldő gép csak az üzenetküldés után továbbítja a vezérlőt.



2. ábra: Gyűrű topológia

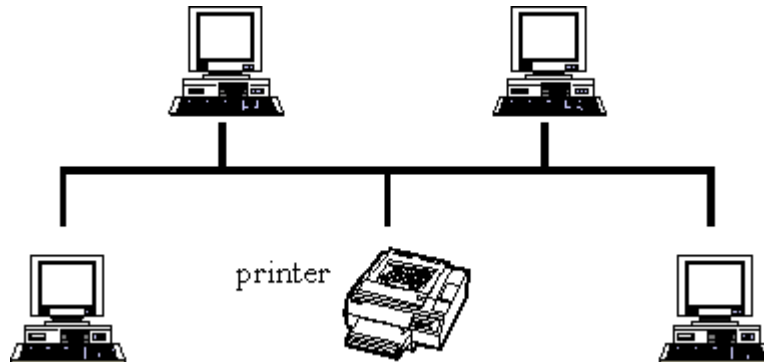
Sín (busz) topológia

A sín topológia valószínűleg a legegyszerűbb hálózati elrendezés. Ez az elrendezés egyetlen, busznak nevezett átviteli közeget használ. A buszon lévő mindegyik számítógépnek egyedi címe van, ez azonosítja a hálózaton.

Egy busz topológiájú hálózat esetén a számítógépeket az esetek többségében koaxiális kábellel csatlakoztatják. Nem egyetlen hosszú kábel, hanem sok rövid szakaszból áll, amelyeket T-csatlakozók segítségével kötnek össze. Ezenkívül a T-csatlakozók lehetővé teszik a kábel leágazását, hogy más számítógépek is csatlakozhassanak a hálózathoz. Egy speciális hardverelemet kell használni a kábel mindkét végének lezárásához, hogy ne verődjön vissza a buszon végighaladó jel, azaz ne jelenjen meg ismételt adatként. Ahogy az adat végighalad a buszon, mindegyik számítógép megvizsgálja, hogy eldöntse, melyik számítógépnek szól az üzenet. Az adat vizsgálata után a számítógép vagy fogadja az adatot, vagy figyelmen kívül hagyja, ha az nem neki szól.

A busz topológiával az a probléma, hogy ha a buszkábel bárhol megszakad, a szakadás egyik oldalán lévő számítógépek nem csak az összeköttetést veszítik el a másik oldalon lévőkkel, hanem a szakadás következtében mindkét oldalon megszűnik a lezárás. A lezárás megszűnésének hatására a jel visszaverődik és meghamisítja a buszon lévő adatokat.

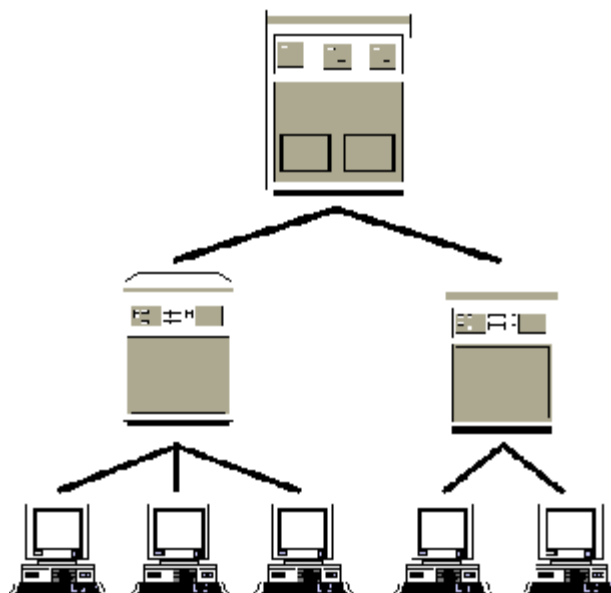
Ha úgy döntünk, hogy busz topológiájú hálózatot alakítunk ki, akkor korlátozott a buszhoz köthető gépek száma. Ez amiatt van, mert ahogy a jel a kábelen halad, egyre inkább gyengébb lesz. Ezt azzal magyarázhatjuk, hogy minden egyes hoszt felfűzésével a T-dugók illesztésénél kábelszakadások keletkeznek. Ha sok hosztot fűzünk fel egy szegmensre, akkor sok szakadás keletkezik, ezáltal megnő az ellenállás és gyengébb lesz a jel. Ez okból kifolyólag, ha több számítógépet csatlakoztatunk a hálózathoz, akkor használnunk kell egy jelerősítőnek (repeater) nevezett speciális hálózati eszközt, amely a busz mentén meghatározott helyeken felerősíti a jeleket. Előnye az egyszerűsége és olcsósága, hátránya viszont, hogy érzékeny a kábelhibákra.



3. ábra: Busz topológia

Fa topológia

A fahálózat jellemzője a központi, kiemelt szerepkört betöltő számítógép. A központi gép ún. közvetítő gépekkel vagy munkaállomásokkal van összekötve. Van egy gyökér, amelyre rákapcsolódnak a kisebb központok. Azután ezekre a kisebb központokra kapcsolódnak a kliens gépek vagy még kisebb szerverek. Tehát a munkaállomások hierarchikus rendben kapcsolódnak egy vagy több másik munkaállomáshoz. Egy-egy ilyen ágat alhálózatként is nevezünk. Minden összekötött gép között csak egyetlen út van. Előnye a kis kábelezési költség, valamint, hogy nagyobb hálózatok is kialakíthatók. Hátránya viszont, hogy egy kábel kiesése egy egész alhálózatot tönkreteszhet.



4. ábra: Fa topológia

A következő táblázat az egyes topológiák összehasonlítását tartalmazza, olyan kritériumok alapján, mint a megbízhatóság, összetettség, rugalmasság, bővíthetőség, költségek, kapacitás.

	Csillag	Sín	Osztott sín	Gyűrű
Megbízhatóság	Közepes	Nagy	Nagy	Közepes
Összetettség	Alacsony	Közepes	Közepes	Alacsony
Rugalmasság	Közepes	Nagy	Nagy	Közepes
Bővíthetőség	Közepes	Közepes	Nagy	Közepes
Költségek	Nagy	Közepes	Közepes	Közepes
Kapacitás	Kicsi	Közepes	Nagy	Nagy

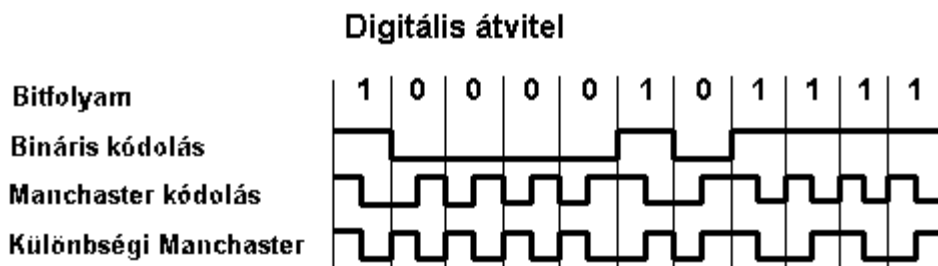
Átviteli módszer alapján

Az átviteli mód arról ad felvilágosítást, hogy miként alkalmazzák összeköttetésre a fizikai közeget. Ahhoz, hogy a hálózatban kapcsolt számítógépek egymással kommunikálni tudjanak, az adatátviteli közegen információt kell cserélniük. Az információ továbbítása valamiféle jelek átviteli közegen való áthaladását jelenti. Amit az egyik ad, azt a másik képes venni. Lehetséges módok: 1. alapsávú vagy digitális, 2. szélessávú vagy analóg, 3. aszinkron kommunikáció és 4. a szinkron kommunikáció.

Alapsávú vagy digitális átvitel

A helyi hálózatok közül általában azokat nevezik alapsávúnak, amelyekben az átvivőközegen haladó jel digitális. A jel a bináris értékeket képviseli. A vonalon egymás után jól meghatározott időközönként jelennek meg az elektromos vagy fény impulzusok, amelyek az üzenet bináris kódjának, 0 és 1 értékek felelnek meg.

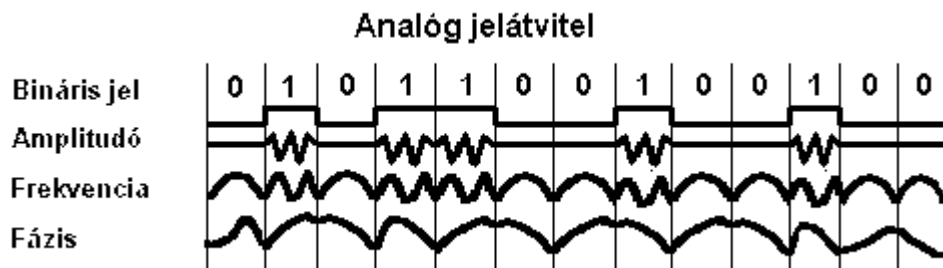
Amikor a vonalon nem történik adás, az elektromosan teljesen csendes, optikailag pedig, ami ugyanennek megfelelő, teljesen sötét. A jel haladása során különféle zavaró hatások következtében információt veszíthet, emiatt van szükség az adatátviteli közegbe iktatott jelismétlőkre, amelyek által a jelek visszanyerik eredeti értéküket. Az alapsávú hálózatok az átvivőközegen más hálózattal nem osztoznak



6. ábra: Digitális átvitel

Szélessávú vagy analóg átvitel

A vonalon állandóan végighalad egy elektromágneses vivőhullám, amelynek három alapvető tulajdonsága valamelyikét módosítják a továbbítandó üzenetnek megfelelően. Ez a három alapvető tulajdonság: a frekvencia, az amplitúdó és a fázis. (frekvencia: az elektromágneses hullám másodpercenkénti rezgésszáma, mértékegysége a Hertz (Hz), amplitúdó: a rezgőmozgás nyugalmi helyzete és maximális kitérése közötti távolság, fázis: két hullám akkor van fázisban, ha a frekvenciáik megegyeznek, és egymásnak megfelelő pontjaik egy adott pillanatban ugyanabban a rezgési állapotban vannak.) A jellegzetes szélessávú hálózatok átvivőközegén frekvenciaosztással több, egymástól független csatornát képeznek, amely csatornák mindegyikén önálló adathálózat alakítható ki. E hálózatok egyesíthetők egyetlen hálózattá, vagy használhatók külön-külön is, egy-egy sajátos feladat ellátására.



7. ábra: Analóg átvitel

A hírközlő csatornák számának megfelelően beszélhetünk egycsatornás (egysávú) és többcsatornás (többsávú) hálózatokról. Az alapsávú hálózatok jellemző módon egycsatornásak, a szélessávú hálózatok pedig általában többcsatornásak.

Az üzenetek (csomagok) lehetnek bit-, oktett- vagy karakterszervezésűek. A bitszervezésű üzenet tetszőleges számú bitből áll, az oktett-szervezésű bitben mért üzenet hossza mindig 8 (esetleg 16) egész számú többszöröse, a karakterszervezésű üzenet pedig valamely adott karakterkészlet (pl. ASCII) karaktereiből épül fel.

Aszinkron kommunikáció

Más néven start-stop mód a karakterszervezésű üzenetek jellegzetes átviteli módja. Az összeköttetésben álló két állomás közül bármelyik adásba kezdhet, nincs meghatározva az időintervallum, de ilyenkor ezt a vevő tudomására kell hozni.

Szinkron mód

A bitszervezésű üzenetek jellegzetes átvitel módja. Szinkron kommunikáció esetén a két kapcsolatba lépő állomás egyezteteli az adatátviteli sebességet, vagyis azt, hogy másodpercenként hány darab jel kerüljön továbbításra, valamint meddig történhet az adás. Az adó elkezd az adást, és a rendelkezésre álló idő alatt átküldi a továbbítandó üzenetnek bitjeit. A vevő folyamatosan veszi a biteket és összeállítja azokat számára feldolgozható byte-okká.

A TCP/IP protokoll család

A Linux és más operációs rendszerek alapvetően a TCP/IP protokollt használják. Pontosabban szólva, ez nem is egy egyedülálló hálózati protokoll, sokkal inkább egy különféle szolgáltatásokat nyújtó hálózati protokollcsalád. A 30.1. táblázat - A TCP/IP protokollcsalád különféle protokolljai ábrán felsorolt protokollok két gép közötti TCP/IP alapú adatcserére szolgálnak. A TCP/IP protokollcsalád segítségével összekapcsolt hálózatok egy világméretű hálózatot alkotnak, az „Internetet”.

Az RFC a *Request for Comments* (megjegyzések kérése) kifejezés rövidítése. Az RFC-k a különféle internetes protokollokat, illetve az operációs rendszerek és alkalmazások számára a megvalósítási eljárásaikat leíró dokumentumok. Az RFC dokumentumok leírják az internetes protokollok beállításának módját is. A protokollokkal kapcsolatos ismeretek bővítése érdekében érdemes elolvasni a megfelelő RFC dokumentumokat. Ezek online változata a <http://www.ietf.org/rfc.html> címen található.

A TCP/IP protokollcsalád különféle protokolljai

Protokoll	Leírás
TCP	Transmission Control Protocol: Kapcsolatorientált, biztonságos protokoll. Az alkalmazás az adatokat egy folyamként küldi ki, amelyet azután az operációs rendszer a megfelelő formára alakít. Az adat a célgépen futó megfelelő alkalmazáshoz mindig az eredetileg elküldött adatfolyam formájában érkezik meg. A TCP megállapítja, hogy veszték-e el az adatok az átvitel során, illetve hogy nincs-e keveredés. A TCP ott kerül alkalmazásra, ahol az adatok sorrendje fontos.
UDP	User Datagram Protocol: Kapcsolat nélküli, nem biztonságos protokoll. Az adatok az alkalmazás által előállított csomagok formájában kerülnek továbbításra. A fogadó félhez érkező adatok sorrendje nem garantált, adatvesztés is előfordulhat. Az UDP a rekordorientált alkalmazások számára hasznos. Elnye a TCP-vel szemben a kisebb késleltetés.
ICMP	Internet Control Message Protocol: Ez igazából nem a végfelhasználóknak szánt protokoll, hanem egy különleges vezérlési protokoll, amely hibajelentéseket biztosít, illetve képes ellenrizni a TCP/IP-adatátvitelben résztvevő gépek viselkedését. Ezenkívül van egy egyedi visszhang üzemmódja is, amelyet például a ping program használ.
IGMP	Internet Group Management Protocol: Ez a protokoll szabályozza a gép viselkedését IP multicast (többesszórás) használata közben.

A 30.1. ábra - A TCP/IP egyszersített rétegmodellje jól mutatja, hogy az adatcsere több szinten, rétegben zajlik: A tényleges hálózati réteg az IP (Internet protokoll) alapú, nem biztonságos adatátvitel. A TCP (transmission control protocol) az IP protokollra épül rá, és azt használva valósít meg biztonságos adatátvitelt. Az IP-réteg maga is ráépül egy legalsó, hardverszintű protokollra, mint amilyen például az Ethernet.

A TCP/IP egyszersített rétegmodellje



Az ábrán minden rétegre egy vagy két példát láthatunk. A rétegek az *absztrakciós szinteknek* megfelelően vannak elrendezve. A legalsó szinten a hardverhez közeli réteg található. A legfelső réteg ezzel szemben már szinte teljesen elvonatkoztat a hardversajátosságoktól. Minden rétegnek megvan a saját speciális funkciója. Az egyes rétegek szerepe általában kiderül a leírásukból. Az adatkapcsolati és a fizikai rétegek jelentik a használt fizikai hálózatot (például Ethernet).

A hardverközeli protokollok szinte mindegyike csomagalapú megoldást alkalmaz. Az átvinni kívánt adatok *csomagokba* szerveződnek, mivel egyszerre nem küldhet el mind. Egy TCP/IP csomag mérete maximum 64 kilobájt lehet. A csomagok általában ennél azonban sokkal kisebbek, mert a hálózati hardver korlátozó tényezőt jelent. Az adatcsomag maximális mérete például egy Ethernet-szegmensben 1500 bájt. A TCP/IP-csomag mérete maximum ekkora lehet, ha az adatok Ethernet-hálózaton keresztül kerülnek továbbításra. Ha több adatot szeretnénk továbbítani, akkor az operációs rendszernek több adatcsomagot kell elküldenie.

Hogy a rétegek elvégezhesék a nekik szánt feladatot, minden réteg számára kiegészítő információt kell elmenteni az adatcsomagokba. Ez az információ a csomag *fejlécében* található. Minden réteg egy rövid adatblokkot, ún. protokollfejléceket fűz a csomagok elejére. A 30.2. ábra - TCP/IP Ethernet-csomag ábra egy TCP/IP adatcsomag továbbítására mutat példát Ethernet-kábelon. Az ellenőrző összeg nem a csomag elején, hanem a végén található. Ez leegyszersíti a hálózati hardver dolgát.

TCP/IP Ethernet-csomag



Amikor egy alkalmazás adatokat küld a hálózaton keresztül, az adatok a fizikai réteg kivételével olyan rétegeken haladnak keresztül, amelyeket a Linux-kernel tartalmaz. Minden réteg felelős azért, hogy az adatokat elkészítse a következő réteg számára. Az adatok tényleges elküldéséért a legalacsonyabb réteg felelős. Adatok fogadása esetén az egész folyamat fordítva zajlik le. A rétegek olyanok, mint egy hagyma: az egyes rétegekben a protokollfejlécek leválasztásra kerülnek a szállított adatokról. Végül a szállítási réteg felelős azért, hogy a célgép alkalmazásai számára felhasználható adatokat állítson el. Mindez azt jelenti, hogy egy réteg csak a közvetlenül felette és alatta lévő rétegekkel kommunikálhat.

Az alkalmazásoknak mindegy, hogy az adat egy 100 megabit/másodperc sebesség FDDI hálózaton, vagy egy 56 kilobit/másodperces modemén keresztül érkezik. Az adatvonalnak is mindegy, hogy milyen adatokat továbbít, feltéve, hogy azok formátuma megfelel.

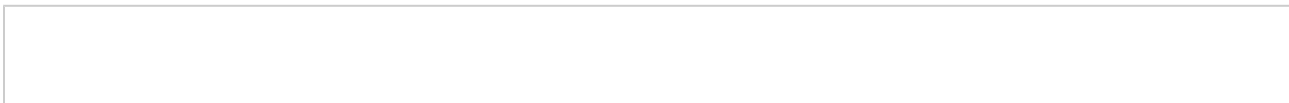
IP-címek és útválasztás

Az alábbi szakaszban csak az IPv4 hálózatokkal foglalkozunk. Az IPv4-et felváltó IPv6 protokollal kapcsolatos további információ: [30.2. szakasz - IPv6 – az Internet következő generációja](#).

IP-címek

Az Internet minden egyes számítógépe saját 32 bites címmel rendelkezik. Ezt a 32 bitet (azaz 4 bájtot) általában a következő példa második sorában látható módon írjuk: [30.1. példa - IP-címek leírása](#).

30.1. példa - IP-címek leírása



Decimális formában a négy bájtot tízes számrendszerben ábrázoljuk, pontokkal elválasztva. IP-címe egy gépnek, illetve egy hálózati csatlónak lehet (és kell lennie). Ez az IP-cím a világon sehol máshol nem használható. A szabály alól vannak bizonyos kivételek, de a következő részekben ezek jelensége elhanyagolható.

Az IP-címekben látható pontok egy hierarchikus rendszerre utalnak. Az 1990-es évekig az IP-címek szigorúan osztályokba voltak sorolva. Ez a rendszer azonban túlságosan merevnek bizonyultak, ezért beszüntették a használatát. Napjainkban ezért az *osztály nélküli útválasztást* (CIDR, classless interdomain routing) használjuk.

Hálózati maszkok és útválasztás

A hálózati maszk az alhálózat címtartományát adja meg. Az azonos alhálózaton lévő gépek közvetlenül el tudják egymást érni. A különböző alhálózatban lévő gépeknek szükségük van az átjáró címére, amely kezeli az alhálózat és a világ többi része közötti forgalmat. Annak ellenzéséhez, hogy két IP-cím ugyanabban az alhálózatban van-e, egyszerűen hozzá „ÉS” kapcsolatba a két címet és a hálózati maszkot. Ha az eredmények megegyeznek, akkor mindkét IP-cím azonos helyi hálózaton található. Ha az eredmények különböznek, akkor az IP-cím távoli, és a távoli csatlakozó csak átjárón keresztül érhető el.

A hálózati maszk működésének megértéséhez tekintse meg a következő részt: [30.2. példa - IP-címek és hálózati maszkok összekapcsolása](#). A hálózati maszk 32 bitből áll, amely mutatja, hogy az IP-cím mekkora része tartozik a hálózathoz. Az 1-es bitek jelzik, hogy az IP-cím megfelelő biteje a hálózathoz tartozik. A 0-ás bitek az alhálózatban lévő biteket jelzik. Ez azt jelenti, hogy minél több 1-es bit van, annál kisebb az alhálózat. Mivel a hálózati maszk mindig számos egymást követő 1-es bitből áll, a hálózati maszkban lévő bitek egyszerűen megszámlálhatók. [30.2. példa - IP-címek és hálózati maszkok összekapcsolása](#) esetében az első 24 bitet tartalmazó hálózat a következőképp is leírható: 192.168.0.0/24.

30.2. példa - IP-címek és hálózati maszkok összekapcsolása

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:        11000000 10101000 00000000 00000000
In the decimal system:      192.      168.      0.      0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:        11010101 10111111 00001111 00000000
In the decimal system:      213.      95.      15.      0
```

Egy másik példa: az ugyanarra az Ethernet-kábelre csatlakozó gépek rendszerint egy alhálózatban találhatók és közvetlenül elérhetők. Ha az Ethernet-hálózatot kapcsolók (switch) vagy hidak (bridge) osztják fel, ezek a gépek még mindig közvetlenül elérhetők.

A helyi alhálózaton kívüli IP-címek csak akkor érhetőek el, ha egy átjáró be van állítva a célhálózathoz. A legáltalánosabb esetben csak egy átjáró van, amely az összes külső forgalmat kezeli. Azonban a különböző alhálózatokhoz több átjáró is beállítható.

Átjáró megadása esetén az IP-csomagok a megfelelő átjárón keresztül továbbítódnak. Az átjárók ugyanúgy továbbítják a csomagokat –géptől gépig –, amíg az eléri a címzett gépet vagy a csomag TTL-je (time to live – élettartam) le nem jár.

Speciális címek

Címtípus	Leírás
Hálózati alapcím	Ez a hálózati maszk és bármely hálózati cím ÉS kapcsolata, ahogy az a következő példa Eredmények részben látható: 30.2. példa - IP-címek és hálózati maszkok összekapcsolása . Ez a cím nem rendelhető egy géphez sem.
Nyilvános (broadcast) cím	Ez lényegében azt jelenti, hogy az „Alhálózat minden gépe.” Ezt úgy állítjuk el, hogy a hálózati maszkot bináris formátumra alakítjuk és a hálózati alapcímmel logikai VAGY kapcsoljuk hozzá. A fenti példa eredménye így 192.168.0.255. Ez a cím egy géphez sem rendelhető.
Helyi gép	A 127.0.0.1 cím szigorúan a „loopback eszköz” számára van kijelölve. Ezen a címen keresztül kapcsolat létesíthető a saját géppel.

Mivel az IP-címek az egész világon egyediek, nem szabad önkényesen kitalált címekkel csatlakozni a világhálóra. Három címtartomány van fenntartva saját, zárt célokra szánt, IP alapú hálózat kialakítására. Bizonyos trükkök alkalmazása nélkül ezekkel a címekkel nem lehetséges az Internet felé kapcsolatot létesíteni, hiszen ezek a címek nem kerülnek továbbításra az Interneten. Ezeket a címtartományokat az RFC 1597 definiálja és a 30.3. táblázat - IP-címtartományok privát felhasználásra mutatja be ket.

IP-címtartományok privát felhasználásra

Hálózat/hálózati maszk	Tartomány
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x – 172.31.x.x

192.168.0.0/255.255.0.0	192.168.x.x
-------------------------	-------------

Alkalmazási réteg

Az alkalmazási réteg (*application layer*) a hét réteg OSI modell legfelső rétege. A felső szintű protokollok kezelését, a megjelenítést, a kódolást és a párbeszéd vezérlését végzi el. A TCP/IP protokollkészlet minden alkalmazási vonatkozású feladatát egyetlen rétegbe szít, ezzel biztosítja az adatok megfelelő csomagolását, mielőtt azok a következő réteghez kerülnének. A TCP/IP az internet és a szállítási rétegre vonatkozó elírásokat is tartalmazza, ahogy a gyakrabban használt alkalmazásokra vonatkozókat is. A csatorna adategységei a APDU rövidítéssel jelölt Application Protocol Data Unit-ok. A réteg széles körben igényelt szolgáltatásokat tartalmaz. Pl.: fájlok gépek közötti másolása, elektronikus levelezés vagy virtuális terminálok elvén oldják meg az ebben a rétegben használt protokollok által összekötött alkalmazások működését. Az alkalmazási réteg virtuális terminál szoftvereket tartalmaz.

Protokolljai

- FTP
- TFTP
- HTTP
- IMAP
- NFS
- POP3
- SMTP
- SPDY
- Telnet
- SNMP
- DNS
- DHCP

Továbbiak: https://hu.wikipedia.org/wiki/TCP_%C3%A9s_UDP_portsz%C3%A1mok_list%C3%A1ja

A HTTP protokoll

A HTTP (HyperText Transfer Protocol) egy információátviteli protokoll elosztott, kollaboratív, hipermédiás, információs rendszerekhez. RFC 2616

A HTTP fejlesztését a World Wide Web Consortium és az Internet Engineering Task Force koordinálta RFC-k formájában. Az 1999-ben kiadott RFC 2616 definiálja a HTTP/1.1-et, amit 2015 végére leváltott a HTTP/2.0-ás verzió, amit az RFC 7540 definiál. Hivatalosan ez a legújabb protokoll.

A HTTP egy kérés-válasz alapú protokoll kliens és szerver között. A HTTP-klienseket a „user agent” gyjtnévvel is szokták illetni. A user agent jellemzően, de nem feltétlenül webböngésző.

A HTTP a TCP/IP réteg felett helyezkedik el. A HTTP implementálható más megbízható szállítási réteg felett is, akár az interneten, akár más hálózaton. Kizárólagosan TCP protokollt használ, mivel az adatvesztés nem megengedhető.

Kérés (request)

Egy HTTP kérés első sora mindig „METÓDUS ERFORRÁS VERZIÓ” alakú, például így:

```
GET /images/logo.gif HTTP/1.1
```

Ezt a sort követheti tetszőleges számú *header* sor „HEADER: ÉRTÉK” alakban, például így:

```
Host: origo.hu
Connection: close
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9)
Gecko/2008052906 Firefox/3.0
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7
Cache-Control: no-cache
Accept-Language: de,en;q=0.7,en-us;q=0.3
```

- A fenti példában a User-Agent sor a kliens által használt webböngésző típusát jelöli.
- Az Accept-Charset sor jelzi, hogy milyen karakter kódolásban várja a szervert a választ.
- Az Accept-Language hasonlóan a válaszban elfogadott nyelvet jelöli.
- A Cache-Control határozza meg, hogy a kliens és szerver közti útvonalon kötelezően használjanak-e cache-elést, vagy sem.

A *header* sorok végét egy üres sor jelzi, melyet az opcionális üzenettest követ. A sorokat a CRLF (azaz kocsis vissza + soremelés) karakterpárral kell elválasztani. A *header*ek végét jelző üres sor csak ezt a két karaktert tartalmazhatja, nem lehet benne szóköz és tabulátor sem.

Metódusok

HTTP protokoll nyolcféle metódust definiál. A metódusok (más szóval *verbek*) a megadott erőforráson végzendő műveletet határozzák meg.

verb	jelentés
HEAD	Ugyanazt adja vissza, mint a GET, csak magát az üzenettestet hagyja ki a válaszból.
GET	A megadott erőforrás letöltését kezdeményezi. Ez messze a leggyakrabban használt metódus.
POST	Feldolgozandó adatot küld fel a szerverre. Például HTML lap tartalmát. Az adatot az üzenettest tartalmazza.
PUT	Feltölti a megadott erőforrást.
DELETE	Törli a megadott erőforrást.
TRACE	Visszaküldi a kapott kérést. Ez akkor hasznos, ha a kliens oldal arra kíváncsi, hogy a köztes gépek változtatnak-e, illetve mit változtatnak a kérésen.
OPTIONS	Visszaadja a szerver által támogatott HTTP metódusok listáját.
CONNECT	Átalakítja a kérést transzparens TCP/IP tunnellé. Ezt a metódust jellemzően SSL kommunikáció megvalósításához használják.

Biztonságos metódusok

Biztonságosnak azokat a metódusokat nevezzük, amelyek csak információ lehívására szolgálnak és elvben nem változtatják meg a szerver állapotát. Más szóval mellékhatás nélküliek. Ilyenek például a HEAD, a GET, az OPTIONS és a TRACE. Fontos megjegyezni, hogy a gyakorlatban lehetnek a biztonságos metódusoknak is szerveroldali mellékhatásai. Elfordulhat például, hogy egy GET kérés hatására a szerver cache-elésbe kezd. Ennél veszélyesebb az, amikor a szerver egy egyszeri hiperlinkkel mutató GET kérés hatására végez módosítást vagy törlést az adatbázisban. Ez a gyakorlat nem ajánlott, mert problémákat okozhat cache-el, keres vagy egyéb automatizált klienseknél, mert ezek nem kívánt változásokat okozhatnak a szerveren az ilyen jellegű GET-eknél.

Idempotens metódusok

Idempotensnek azokat a metódusokat nevezzük, melyeknek többszöri végrehajtása ugyanazt a hatást váltja ki, mint az egyszeri. Ilyenek például a *PUT* és a *DELETE*. Minden biztonságos metódus (például HEAD, GET, OPTIONS és TRACE) értelemszerűen idempotens is. Az RFC szerint az idempotens metódusoknál a kliens (leggyakrabban webböngésző) következmény nélkül újrapróbálkozhat, ha a kérés sikertelen volt. Ez arra jó, hogy ha a szerver túl lassan vagy hibásan válaszol, akkor a böngésző felhasználói beavatkozás nélkül újrapróbálhatja az oldal letöltését.

Fontos azonban tudni, hogy a szabványban definiált idempotencia nem nyújt automatikus védelmet a szerveroldali változásoktól. Minden további nélkül írható olyan webalkalmazás, amely GET kérés hatására adatbázis-módosítást (sql update) vagy -beszúrás (sql insert) hajt végre, amelyek nyilvánvalóan szerveroldali változást okoznak.

Az ilyen GET használat és a kliens automatikus újrapróbálkozásának összehasonlítása nemkívánatos eredményeket okozhat, ezért a GET

használata tranzakciók esetében kerülendő. Tanácsos követni az RFC-ben definált szabványt, és a GET metódust csak adatok lehívására használni.

Feltételes kérés

A feltételes kérésre azért van szükség, hogy meggyorsítsuk a HTTP kommunikációt a **cache**-elés segítségével. Mivel a web-szerverek és böngészek képesek az oldalak, fájlok, képek ideiglenes tárolására, így nincs szükség ismételt letöltésre. Ezt a lekérést a feltételes (conditional) GET metódus végzi, melyet a kliens akkor küld a szerver irányában, ha azt már legalább egyszer meglátogatta (az els lekérés során csak általános GET lekérés történik).

A feltételes GET kérést két 'header' sor jelöli:

```
GET /pelda.html HTTP/1.1
(...)
If-Modified-Since: Sat, 22 Oct 2001 19:43:31 GMT
(...)
If-None-Match: „kód”
```

- Az If-Modified-Since-el a kliens lekérdezi a szervert, hogy a kért dokumentum módosult-e a megadott dátum óta.
- Az If-None-Match a szerver által küldött dokumentum hash kódjára hivatkozik.

Válasz (response)

A HTTP válasz els sora a státuszsor, amely „VERZIÓ STÁTUSZKÓD INDOKLÁS” alakú. A státuszkód (angolul *status code*) egy három számjegy szám, az indoklás (angolul *reason phrase*) egy üzenet valamilyen emberi nyelven. Az elbbit inkább gépi, az utóbbit inkább emberi feldolgozásra szánták. Például:

```
HTTP/1.1 200 OK
```

vagy

```
HTTP/1.1 404 Not Found
```

A szöveges üzenetekre a szabvány csak javaslatokat tartalmaz. A szerver küldhet lokalizált üzeneteket is:

```
HTTP/1.1 404 Nincs meg.
```

Státuszkódok

A státuszkódok jelentését az RFC 2616 tartalmazza részletesen, az alábbi lista egy áttekint osztályozást ad a kezd számjegy alapján:

- 1xx: Informatív – Kérés megkapva.
 - Pl.: 100 – Folytatás, 101 – Protokoll váltás

- 2xx: Siker – A kérés megérkezett; értelmezve, elfogadva.
 - Pl.: 200 – OK, 202 – Elfogadva, 203 – Nem autoritativ információ
- 3xx: Átirányítás – A kérés megválaszolásához további mveletre van szükség.
 - Pl.: 301 – Ideiglenesen elköltözött, 305 – Használjon proxy-t
- 4xx: Kliens hiba – A kérés szintaktikailag hibás vagy nem teljesíthet.
 - Pl.: 403 – Nem engedélyezett, 404 – Nem található
- 5xx: Szerver hiba – A szerver nem tudta teljesíteni az egyébként helyes kérést.
 - Pl.: 503 – Szolgáltatás nem elérhet, 505 – Nem támogatott HTTP verzió

Ha a státuszkód hibára utal, akkor a kliens megjelenítheti a hibaüzenetet, hogy tájékoztassa a felhasználót a hiba természetéről. A szabvány megengedi azt is, hogy a kliens maga interpretálja a státuszkódot és az alapján saját üzenetet generáljon a felhasználónak, de ez zavaró lehet. A szabvány szerint a státuszkódot szánják gépi feldolgozásra, és a „reason phrase” való emberi fogyasztásra. Használhatóak egyedi státuszkódok is, mert a kliens ismeretlen kód esetén az els számjegy alapján már tudja osztályozni a választ.

Header sorok

A státuszsor után *header* sorok következhetnek a HTTP kérésnél látott módon „HEADERNÉV: ÉRTÉK” alakban. Például így:

```
Server: Apache
Date: Sat, 24 Mar 2012 16:49:31 GMT
Content-type: text/html
Pragma: no-cache
Connection: close
```

- A Server magán a szerveren futó kiszolgáló szoftvert azonosítja.
- A Date az elküldött válasz dátumát tartalmazza.
- A Content-type: a válaszban (body-ban) elküldött szöveg típusát tartalmazza.
- Pragma: a kliens oldalon futó böngész nem fogja cache-elni a lekért adatokat.

A *header* sorokat itt is üres sor zárja, melyet az opcionális üzenettest követ.

A kliens elssorban a státuszkód, másodsorban a header sorok tartalma alapján kezeli a választ.

Feltételes válasz

A feltételes válasz kétféle lehet, attól függően, hogy a kért adat szerepel-e már a kliens gyorsítótárjában.

Ha a kliens elsször látogatja az oldalt, akkor a következ válasz érkezik:

```
HTTP/1.1 200 OK
(...)
Cache-Control: max-age=21600
Last-Modified: Wed, 01 Sep 2009 13:24:52 GMT
Etag: "4586bdc8"
```

A Cache-Control megadja a kliensnek, hogy mennyi ideig cache-elje (tárolja) a dokumentumot.

- A Last-Modified megadja, hogy mikor lett utoljára módosítva a dokumentum.
- Az Etag egy egyedi azonosító (hash) a dokumentumhoz.

Ezt a választ követően a kliens elmenti a dokumentumot a *max-age* paraméterben megadott ideig. Legközelebb, amikor a kliens ismét meglátogatja az oldalt, már a fentebb leírt feltételes kéréssel hivatkozik a dokumentumra. Ha a dokumentum nem módosult, akkor a következ válasz érkezik a szerverről:

```
HTTP/1.1 304 Not Modified (...)  
Keep-Alive: timeout=2, max=99  
Etag: "4586bdc8"  
Cache-Control: max-age=21600
```

- A szerver látja, hogy még nem módosult a dokumentum az utolsó lekérés óta, ezért egy 304-es kódot küld a kliensnek.
- Beállítja ismételten a dokumentumhoz tartozó értékeket.

A szerver csak akkor fogja elküldeni a kliensnek a kért dokumentumot, ha az módosult a cache-elés óta.

Hatékonyságnövelés

A HTTP/0.9 és 1.0 verziókban a kapcsolat egy kérdés-válasz után lezáródik. A HTTP/1.1 verzióban bevezettek egy mechanizmust a kapcsolat életben tartására, így a kapcsolat újrafelhasználható további kérésekhez. A kapcsolat életben tartását hívják idegen szóval perzisztenciának, az ilyen kapcsolatot pedig a perzisztens jelzvel illetik. Ez az újítás gyorsíthatja a kommunikációt, mert a kliensnek nem kell újratárgyalnia a TCP kapcsolatot minden egyes kérésnél.

Egy másik teljesítménynövelő újítás a HTTP/1.1 verzióban az ún. *chunked transfer encoding*, mellyel a perzisztens kapcsolat felett adatfolyam (*stream*) továbbítható, a kevésbé gazdaságos pufferek helyett. A *HTTP pipelining* segítségével pedig a kliens elküldhet több kérést is egymás után anélkül, hogy megvárná a választ. További hatékonyságoptimalizáló újítás a *byte serving*, ami azt jelenti, hogy a kért erőforrásnak csak a kliens által kijelölt darabját (byte range) küldi el a szerver.

Munkamenet (session)

A HTTP egy állapot nélküli protokoll. Az állapot nélküli protokollok elnye, hogy a szervernek nem kell nyilvántartania felhasználói információkat az egyes kérések kiszolgálása között. A HTTP eredetileg nem arra készült, hogy felhasználók jelentkezzenek be rajta keresztül szerverekre és ott munkamenetet (idegen szóval session-t) indítsanak. Történetileg azonban úgy alakult, hogy a HTTP terjedt el széles körben más, felhasználói bejelentkezést támogató protokollok helyett, ami arra kényszerítette a webfejlesztőket, hogy a HTTP-t mintegy megerszakolva, kerülutakon járva tárolják a felhasználók munkamenetállapotait. Egy tipikus megoldás cookie-kban tárolni a felhasználói állapotot. Egyéb módszerek még a rejtett változók (például `<input type=hidden name=session_id value="1956">`) vagy az URL-ben kódolt paraméterek (például `/index.php?userid=3`) használata illetve a szerveroldali állapotmegrzés. A legbiztonságosabb megoldás vélhetően a szerveroldali állapotmegrzés, mert az az egyetlen, amelyet nem tudnak „megpiszkálni” rosszindulatú kliensek.

Biztonságos HTTP

Jelenleg két módszer áll rendelkezésre biztonságos http-kapcsolat felépítésére: Az egyik a https URI-séma, a másik pedig a HTTP/1.1 verzióban bevezetett Upgrade header (lásd RFC 2817). Az Upgrade header kliensoldali támogatása jelenleg gyakorlatilag még nem létezik, ezért egyértelműen a https dominál.

A HTTPS URI-séma

A https séma szintaktikailag megegyezik a http-sémával, de jelzi a böngésznek, hogy használni kell az SSL/TSL titkosító réteget az adatforgalom védelme érdekében. Az SSL különösen célszerű a HTTP esetében, mert akkor is nyújt némi védelmet, ha csak a kommunikáció egyik oldala hitelesített (más szóval autentikált). Az internetes HTTP-tranzakciók esetében jellemzően csak a szerveroldal hitelesített.

A HTTP 1.1 Upgrade header

A HTTP 1.1 verzióban bevezették az Upgrade headert. A kommunikáció során a kliens elsősor egy sima titkosítatlan kérést küld, majd később vagy a kliens vagy a szerver kéri (vagy megköveteli) a kapcsolat titkosítását. Jellemzően a szerver követeli meg a titkosítást:

Kliens


```
GET /titkos-cuccok HTTP/1.1
Host: www.valami.com
```

Szerver

```
HTTP/1.1 426 Upgrade
Required Upgrade: TLS/1.0, HTTP/1.1
Connection: Upgrade
```

A 426-os státusz kód jelzi, hogy kötelező a titkosítás, az Upgrade header pedig megadja a támogatott protokollverziókat. Ha a kliens nem támogatja az Upgrade headert és nem ismeri ezt a hibakódot, akkor is tudja az első számjegyből, hogy kliensoldali hibáról van szó.

Kapcsolódó anyagok

<https://hu.wikipedia.org/wiki/Nagysz%C3%A1m%C3%ADt%C3%B3g%C3%A9p>

<https://hu.wikipedia.org/wiki/ARPANET>

https://hu.wikipedia.org/wiki/Bulletin_Board_System

[https://hu.wikipedia.org/wiki/Mosaic_\(b%C3%B6ng%C3%A9sz%C5%91\)](https://hu.wikipedia.org/wiki/Mosaic_(b%C3%B6ng%C3%A9sz%C5%91))

https://hu.wikipedia.org/wiki/Kliens-szerver_architekt%C3%BAra

https://www.novell.com/hu-hu/documentation/sled10/sled_deployment_sp2/html/SLED-deployment/cha.basicnet.html#sec.basicnet.addresses

<https://hu.wikipedia.org/wiki/OSI-modell>

<https://hu.wikipedia.org/wiki/HTTP>

https://hu.wikipedia.org/wiki/TCP_%C3%A9s_UDP_portsz%C3%A1mok_list%C3%A1ja