**Az alkalmazási réteg protokolljai**

**HTTP**

* A HTTP (HyperText Transfer Protocol) egy információátviteli protokoll
* A HTTP fejlesztését a World Wide Web Consortium és az Internet Engineering Task Force koordinálja. Az 1999-ben kiadott HTTP/1.1-et, 2015 végére leváltott a HTTP/2.0-ás verzió. Hivatalosan ez a legújabb protokoll.
* A HTTP egy kérés-válasz alapú protokoll kliens és szerver között. A HTTP-klienseket a „user agent”(ált. böngészők) gyűjtőnévvel is szokták illetni.

|  |  |
| --- | --- |
| 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 űrlap](https://hu.wikipedia.org/w/index.php?title=HTML_%C5%B1rlap&action=edit&redlink=1) 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](https://hu.wikipedia.org/w/index.php?title=SSL&action=edit&redlink=1) kommunikáció megvalósításához használják. |

* Kizárólagosan TCP protokollt használ, mivel az adatveszteség nem megengedhető.
* Ezek a http-hez tartozó alapműveletek

**HTTPS**

* biztonságos http kapcsolatot jelöl.
* Szintaktikailag megegyezik a http sémával, amelyet a HTTP protokollnál használnak, de a https nem önálló protokoll, hanem csak egy URI séma, mely azt jelzi, hogy a HTTP protokollt kell használni
* A https-t a Netscape fejlesztette ki, hogy a webes kommunikáció titkosítható és autentikálható legyen. Ma széles körben használják ezt a rendszert a weben biztonságilag kritikus kommunikációknál, mint amilyenek például a fizetési tranzakciók és a felhasználói jelszavas bejelentkezések.
* Ahhoz, hogy egy webszerver https kéréseket tudjon fogadni, az adminisztrátornak tanúsítványt (angolul public key certificate) kell készítenie.
* A https alapja a kétkulcsos titkosítás. A kulcspár két összetartozó, nagyon nagy (több száz jegyű) szám. A titkosítandó szöveget az egyik szám segítségével, egy nyilvános eljárással rejtjelezzük. Az így kapott üzenet ugyanazzal az eljárással, de a kulcspár másik tagjának segítségével fejthető csak vissza. → Ha a kulcspár egyik tagját titokban tartjuk, a másik kulcsot nyilvánosságra hozzuk, és valaki egy üzenetet a nyilvános kulcsunkkal rejtjelez, akkor azt csak mi fogjuk tudni elolvasni. A titkosítás mindig a nyilvános kulccsal történik. → A nyilvános kulccsal visszafejthetők lesznek azok az üzenetek, amelyeket mi rejtjeleztünk. A visszafejthetőség igazolja, hogy a titkos üzenetet mi írtuk (hiszen a titkosító kulcsot csak mi ismerjük), ezért ezt az eljárást digitális aláírásnak nevezzük. A digitális aláírás tehát mindig titkos kulccsal történő rejtjelezést jelent.
* 2017 januárjában a Chrome webböngésző az olyan weboldalakat, amelyek a titkosított HTTPS protokoll nélkül gyűjtenek adatokat, nem biztonságosként kezdte el megjelölni. Ez a módosítás várhatóan jelentős mértékben növeli majd a HTTPS használati arányát. 2017 februárjában a magyarországi domainnevek kb. 9,4%-ánál alkalmazták a HTTPS protokollt.

**SMTP**

* Az SMTP a Simple Mail Transfer Protocol rövidítése, ami egy kommunikációs protokoll az e-mailek Interneten történő továbbítására.
* Az SMTP egy viszonylag egyszerű, szövegalapú protokoll, ahol egy üzenetnek egy vagy több címzettje is lehet. Könnyen tesztelhetjük az SMTP-t a Telnet program segítségével.
* Az SMTP-t 1982-ben definiálták először
* A Sendmail volt az első levéltovábbító ágens (mail transfer agent) ami megvalósította az SMTP-t. Ezt az SMTP-t használja a közkedvelt Microsoft Exchange Server.
* Az SMTP kezdetben csak a hétbites ASCII karaktereket ismerte, nem tudott mit kezdeni a bináris file-okkal. A felhasználók alkalmi megoldásokat vettek igénybe ilyen esetekben. De mára már kifejlesztették a MIME kódolást, ahol bináris fájlok is „utazhatnak” a levelekben. Ma már minden SMTP kiszolgáló támogatja a 8 bites kiterjesztésű leveleket, ami bináris formában tárolja/küldi az üzeneteket.

**IMAP**

* Az IMAP (Internet Message Access Protocol) alkalmazásrétegbeli protokoll, amely segítségével a leveleinkhez férhetünk hozzá. Mark Crispin fejlesztette ki 1986-ban. A POP3 mellett a legelterjedtebb levéllekérési internetszabvány. A legtöbb modern szerver és kliens is támogatja használatát.
* Előnyei:
  + A levelek nem töltődnek le, a kliens csak cache-eli őket
    - Ezáltal csökken a hálózati forgalom, a kliens háttértárigénye, a levelek bárhol elérhetővé válnak.
  + Állapotinformációk tárolhatóak a kiszolgálón
    - A zászlókon keresztül több információ is tárolhatók a levél állapotáról, például, hogy olvasatlanok, vagy nem, hogy megválaszoltak-e vagy sem.
  + Mappák támogatása
    - Az IMAP4 kliensek képesek létrehozni, átnevezni és törölni postafiókokat, amelyeket a felhasználó általában mappáknak lát. Megosztott és nyilvános mappákat is lehetséges létrehozni.
  + Szerveroldali keresések támogatása
    - A kliensek kérhetik a kiszolgálót, hogy keressen a postafiókban tárolt levelek között. Így elkerülhető az összes levél letöltése.

**POP**

* A Post Office Protocol alkalmazás szintű protokoll, melynek segítségével az e-mail kliensek egy meglévő TCP/IP kapcsolaton keresztül letölthetik az elektronikus leveleket a kiszolgálóról.
* A jelenleg használatos harmadik változat (POP3) elődjei a POP, illetve POP2 változatok.
* A protokollra eredetileg az időszakosan létrejövő TCP/IP kapcsolatok (például dial-up) miatt volt szükség, ugyanis lehetővé teszi a kapcsolódás korlátozott ideje alatt a levelek kezelését a felhasználó gépén, úgy, hogy a levelek összességében akár a szerveren is maradhatnak. A leveleket azután helyben lehet olvasni, szerkeszteni, tárolni stb. A POP3 protokoll kizárólag a levelek letöltésére alkalmas; küldésükre az SMTP protokoll szolgál.
* A POP3 és az IMAP4 protokoll közötti lényeges különbség a levelek kezelési elvében mutatkozik. Az IMAP segítségével a levelező kliens kapcsolódik a kiszolgálóhoz, és közvetlenül azon manipulálja a leveleket. Ugyanakkor a POP3 protokoll esetében a kiszolgáló csak addig tárolja a levelet, amíg a felhasználó le nem tölti onnan. A két módszer közötti filozófiai eltérést kell leginkább szem előtt tartanunk, mivel mindkét protokoll esetében lehetőség van a másikhoz hasonlatos működés megvalósítására: POP3 használatakor a letöltött leveleket nem feltétlenül kell letörölni a szerverről; IMAP esetén pedig a levelek ideiglenesen a felhasználó gépén tárolódnak, és akár offline is olvashatók maradnak.

**DNS**

* A Domain Name System (DNS), azaz a tartománynévrendszer egy hierarchikus, nagymértékben elosztott elnevezési rendszer számítógépek, szolgáltatások, illetve az internetre vagy egy magánhálózatra kötött bármilyen erőforrás számára. A részt vevő entitások számára kiosztott tartománynevekhez (doménekhez) különböző információkat társít. Legfontosabb funkciójaként az emberek számára értelmes tartományneveket a hálózati eszközök számára érthető numerikus azonosítókká „fordítja le”, „oldja fel”, melyek segítségével ezeket az eszközöket meg lehet találni, meg lehet címezni a hálózaton.
* Gyakran használt analógia a tartománynévrendszer magyarázatához, hogy az internet egyfajta telefonkönyve, amiből ki lehet keresni az emberek számára értelmezhető számítógép-állomásnevekhez tartozó IP-címeket. Például a www.example.com tartománynévhez a 192.0.32.10 (IPv4) és a 2620:0:2d0:200::10 (IPv6) címek tartoznak.
* **DNS HIERARCHIA**
  + A DNS fordított fastruktúrájú hierarchiáját egymásba ágyazott tartományok (domének) alkotják, melyek szintjeit ponttal választják el egymástól, fontosságuk pedig jobbról balra haladva egyre csökkenő, pl. sub-b.sub-a.example.com. A fa minden leveléhez vagy csomópontjához nulla vagy több, a hozzá tartozó tartomány információit tároló erőforrásrekord tartozik.

**NSLOOKUP(Name Server LOOKUP)**

* egy parancs amivel információt kérhetünk le egy domainnévről
  + ezt a parancsértelmezőbe kell beírni
* nslookup google.com
  + a parancs után be kell írni azt a domaint aminek az adataira kíváncsiak vagyunk
* válaszként megkapjuk a kiszolgálónk nevét alatta a címét
* üress sorral elválasztva kapunk egy nem mérvadó válasz a domainról kapjuk a nevét és az ipcímét

**DHCP**

* A dinamikus állomáskonfiguráló protokoll (angolul Dynamic Host Configuration Protocol, rövidítve DHCP) egy számítógépes hálózati kommunikációs protokoll.
* Ez a protokoll azt oldja meg, hogy a TCP/IP hálózatra csatlakozó hálózati végpontok (például számítógépek) automatikusan megkapják a hálózat használatához szükséges beállításokat. Ilyen szokott lenni például az IP-cím, hálózati maszk, alapértelmezett átjáró stb.
* A DHCP szerver-kliens alapú protokoll, nagy vonalakban a kliensek által küldött DHCP-kérésekből, és a szerver által adott DHCP-válaszokból áll.
* A DHCP-vel dinamikusan oszthatóak ki IP-címek, tehát a hálózatról lecsatlakozó számítógépek IP-címeit megkapják a hálózatra felcsatlakozó számítógépek, ezért hatékonyabban használhatóak ki a szűkebb címtartományok.
* 3 féle IP-kiosztás lehetséges DHCP-vel:
  + kézi (MAC-cím alapján)
  + automatikus (DHCP-vel kiadható IP-tartomány megadásával)
  + dinamikus (IP-tartomány megadásával, de az IP-címek „újrahasznosításával”)
* A DHCP-forgalom a 67-es és a 68-as UDP portokon zajlik, a kliensek a 67-es portra küldik az üzeneteiket, a szerver pedig a 68-as portra kézbesíti a válaszokat, amelyek szórt üzenetek, vagyis a hálózat minden állomása megkapja őket. A konfiguráció lekérdezését a kliens kezdeményezi egy DHCPDISCOVER (DHCP-felfedezés) üzenettel. Mivel a kliens ekkor még nem rendelkezik hálózati rétegbeli címmel, a DHCP üzenetekben van egy xid nevű mező, ez szolgál a gépek megkülönböztetésére a DHCP-kommunikáció alatt.
* A szerver a feltérképezésre használt DHCPDISCOVER üzenetre a DHCPOFFER (DHCP-ajánlat) üzenettel válaszol, megintcsak üzenetszórással. A kliens csak a saját xid-jével ellátott DHCPOFFER választ dolgozza fel. Az ajánlat nem más, mint a szerver által felkínált IP-cím, amit az ajánlattétel után a kliens még nem használhat azonnal, előtte el kell kérnie. A folyamat következő lépcsője tehát a DHCPREQUEST (DHCP-kérés), amelyben a felajánlott címet a kliens elkéri a szervertől, erre a nagyvonalú szerver megküldi a DHCPACK-et (DHCP-nyugta), ami minden fontos információt tartalmaz ahhoz, hogy a kliens a kézhezvétel után teljes értékű IP-állomásként működhessen. A DHCPACK kézhezvétele után a kliens kötött állapotba (BOUND) kerül, ez a bérleti idő lejártáig érvényben is marad. A folyamat annyira gyors, és olyan kis sávszélességet igényel, hogy az még a ma már elavultnak számító 10 Mb/s-os hálózatokban is szinte észrevehetetlen.