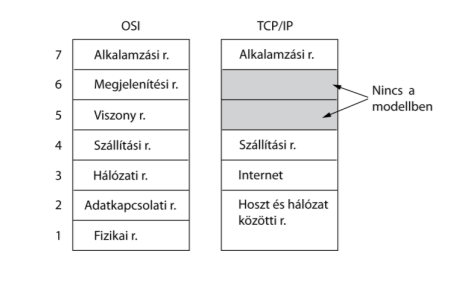
**Hálózatok**

**1. Sorolja fel az OSI és a TCP/IP referenciamodell rétegeit és határozza meg röviden a rétegek feladatait**



* **OSI:**
  + **Fizikai réteg:** 
    - Feladata a bitek továbbítása a kommunikációs csatornán
    - Feszültésgszintek hozzárendelése a logikai értékekhez
    - A fizikai réteg kódolja a kereteket és létrehozza azokat az elektromos, optikai vagy rádióhullám jeleket, amelyek a keret bitjeinek felelnek meg
    - Ezután a jelek egyesével elküldésre kerülnek az átviteli közegen.
    - A célállomás fizikai rétege fogadja ezeket a jeleket a közegen, bitekké alakítja őket, majd a biteket keretként továbbítja az adatkapcsolati rétegnek.
    - Az interfész, mechanikai és elektronikai kérdésekre összpontosít
  + **Adatkapcsolati réteg:**
    - Az átvitel a fizikai rétegnél nem tökéletes, ezért megpróbálja kijavítani
    - Hibaérzékelő és hibajavíró kódokat tartalmaz
    - Keretezés, nyugtázás a csomag vesztések kezelésére
    - Adatforgalom szabalyozasa, hogy a lassu vevot ne arasszak el
    - Mindezen celokra ugynevezett keretekbe agyazza az adatokat
  + **Hálózati réteg:**
    - Feladata a csomag eljutatása a célig (az adatkapcsolási réteg csak egyik hossztól a másikig viszi a csomagot)

Ismernie kell a kommunikációs alhálózatot:

útkeresés

torlódások elkerülése

különböző hálózatok közötti átmenet

Alhálózat működését irányítja

* + - Útkeresés:
      * Statikus: táblázatokban
      * Dinamikus: minden csomag új útvonalat keres
    - Torlódás kezelése
    - Szolgáltatás minőségének biztosítása
    - Különböző hálózatok közti átmenet megoldása (Ipv4 és Ipv6)
  + **Szállítási réteg:**
    - Célja a hatékony, megbízható és gazdaságos adatszállítás biztosítása a forrásgéptől a célgépig függetlenül az alatta elhelyezkedő rétegek típusától.
    - Adatokat fogad a viszony rétegből, feldarabolja és továbbítja a hálózati rétegnek
    - Feladata, hogy az adat hibátlanul megérkezzen a túlsó oldalra, vagy jelezze a hibákat
    - Itt dől el a szolgáltatás típusa: kapcsolat vagy üzenetorientált
  + **Viszonyréteg:**
    - Két hoszt közötti viszony felépítése
    - Párbeszéd kialakítása (handshake), szinkronizáció
  + **Megjelenítési réteg:**
    - Adattípusok közti konverziót hajtja végre
    - encrypt, decrypt, tömörítés
  + **Alkalmazási réteg:**
    - Gyakran használt protokoll sokasága: HTTP, HTTPS, ftp, mail
* **TCP/IP** [**https://www.tutorialspoint.com/The-TCP-IP-Reference-Model**](https://www.tutorialspoint.com/The-TCP-IP-Reference-Model)**:** 
  + **Alkalmazási réteg:**
    - A felhasználó által indított program és a szállítási réteg között teremt kapcsolatot a leggyakrabban használt protokollokat tartalmazza (HTTP, FTP, SMTP)
  + **Szállítási réteg:**
    - Itt dől el, hogy milyen protokollokkal küldi tovább az adatokat (TCP vagy UDP)
  + **Internetréteg – halozati retegnek felel meg:**
    - Az egész architektúrát összefoglalja
    - Bármilyen hálózatba csomagokat tudjon küldeni, és meg is érkezzen a köztes hálózatok típusától és a csomagok sorrendjétől
  + **Host es Halozat réteg:** nem specifikalt, a LAN-tol fug

**-** ez felel meg a fizikai retegnek, a bitek tovabbitasaert felelos

**2. Sorolja fel a TCP protokoll szolgáltatásait, tulajdonságait**

- egy összeköttetés alapú protokoll, nem képes adatszórásra és többesküldésre (multicast)

*- bitfolyamot biztosít* két végpont között, tulajdonságaik: rendezett, hibamentesek, nincs adat kettőzés

- Socketeken kommunikalniak. Socket tartalmazza az Ip cimet es a portot ahol elerik egymast

*- nem örzi meg az üzenet határokat*: küldheti azonnal vagy pufferelheti

*- push függvény*: ha az üzenet azonnali küldi az adatokat, nem puffereli őket, hanem amilyen gyorsan csak lehetséges, továbbítsa

*- sürgős adat*: a protokoll megszakítja az előbbi adatok feldolgozását, ahogy a fogadó is, majd a csomag kezelése után visszatér eredeti csomagok kezeléséhez

- Képes alkalmazkodni az összekapcsolt hálózatok változó paramétereihez.

- A TCP protokoll megvalósítása lehet felhasználói folyamat, könyvtári eljárás vagy az operációs rendszer része.

- A TCP a helyi folyamatoktól kap adatokat, amelyeket szegmensekre bont és IP csomagok segítségével továbbit a hálózatban

A TCP nyújtotta szolgáltatás lényegi jellemzői A TCP lényegi működési jellemzői öt pontban foglalhatók össze:   
**1. Adatfolyam orientált (Stream oriented)**

Az alkalmazás által továbbítani kívánt adatokat bitek – helyesen oktetek – sorozatának tekintjük. Az adatfolyam-szolgáltatás pontosan ugyanazt a bitsorozatot adja át a célgépen futó alkalmazásnak, amelyet az adatokat küldő gép adott át az átviteli szolgáltatásnak.

**2. Virtuális áramköri kapcsolat (Virtual Circuit Connection)**

Telefon kapcsolathoz hasonlítható. Az alkalmazások az operációs rendszerhez fordulnak, kérik az átviteli szolgáltatást. Az operációs rendszer kérésére a végponti protokoll-gépek kapcsolatba lépnek egymással, ellenőrzik, mindkét fél kész-e a kapcsolat létrehozására, majd megállapodnak a részletekben. Ezután a protokoll-gépek értesítik a végpontokon futó két alkalmazást, hogy a kapcsolat létrejött, megkezdhetik a kétirányú adatátvitelt. A virtuális áramkör megléte alatt a két protokoll-gép együttműködése biztosítja, hogy az átvitt adatok hibátlanok legyenek. Csak a helyrehozhatatlan hibákat jelentik az alkalmazásoknak (ekkor azonban a virtuális áramkör megszakad). Azért nevezzük a kapcsolatot virtuális áramkörnek, mert az alkalmazások úgy látják, mintha az egy dedikált, külső körülmények által meg nem zavart kapcsolat lenne. A megbízhatóságot az adatfolyam átviteli protokoll biztosítja.

**3. Pufferelt átvitel (Buffered Transfer)**

Az alkalmazások (közel) tetszőleges mennyiségű adatot adhatnak át átvitelre a szállítási szolgáltatásnak. Az adatokat a protokoll-gép pufferben gyűjti, majd a hatékonyságot szem előtt tartva kisebb-nagyobb csomagokban továbbítja azokat. Szükség esetén akár egy-egy oktet is átvitelre kerül (pl. egy billentyűleütés).   
Az erre szolgáló úgynevezett push mechanizmus kényszeríti a protokollt, hogy a puffer megtelte előtt vigye át az adatot. Ilyenkor a vételi oldalon működő protokoll-gép késleltetés nélkül átadja az adatot az alkalmazásnak.

**4. Strukturálatlan adatfolyam (Unstructured Stream)**

A TCP által kézbesített adat nem strukturált. A szállítási szolgáltatás semmit sem tud az átviendő adatok tartalmáról, azok struktúrájáról. Az alkalmazásoknak kell megegyezniük az adatok szerkezetében, és megérteniük az adatfolyamot, felismerni az abban lévő esetleges adathatárokat.

**5. Egyidejű kétirányú kapcsolat (Full Duplex Connection)**

A TCP kapcsolat egyidejű adatfolyam-átvitelt biztosít mindkét irányban. Az alkalmazások lezárhatják az egyik irányú adatfolyamot, ha kívánják (az ettől kezdve fél-duplex lesz). A kétirányú kapcsolat azért is előnyös, mert az ellenkező irányban haladó adatfolyam vezérlő – szolgálati – üzenetet is továbbíthat. Ez – a piggy-backing-nek nevezett – megoldás csökkenti a hálózati forgalmat

**3. Sorolja fel az UDP protokoll szolgáltatásait, tulajdonságait**

* összeköttetés nélüli protokol
* nem garantálja a csomag célba érkezését
* nem rendelkezik semmilyen visszajelzéssel a felhaszánló fele
* alacsony késleltetéssel rendelkezik
* (8 bájtos fejrész + adat) szegmensbál áll   
  Szolgáltatásai
  + - Interfész az IP protokollhoz (bind kapcsolódik egy portra)
  + - Multiplexeli a csomagokat
  + - UDP nem végez forgalomirányítást, Hibakezelést, Újraküldést rossz szegmens esetén
  + - Alkalmazás – olyan alkalmazások, amelyeknek fontos a pontos időkezelés, csomagforgalom, és hibakezelés
  + - Nem kell kiépíteni a kapcsolatot

- Kapcsolatmentes. A feladó és a címzett nem beszélgetnek: a feladó elküldi a csomagot, a címzett vagy megkapja, vagy nem. De semmilyen formában sem jelez vissza.

- Megbízhatatlan. A csomagokban nincs semmi sorszámozás, nincs semmi elveszés elleni védelem. UDP esetében minden ilyesmit arra az alkalmazásra bíznak, amelyik küldi/fogadja a csomagokat. (CRC ellenőrzés ugyan van, de az csak szótlan csomageldobáshoz vezet.)

- Nem tárol. Ahogy beérkezik egy UDP csomag, az rögtön megy is tovább az alkalmazás réteg felé. Ha az nem veszi át, akkor úgy járt.

- Nem tördel. Tördeljen helyette más. (Ezt mondjuk megteszi az IP réteg.)

- Nincs benne folyamatszabályozás. Azaz nem tud se gyorsítani, se lassítani.

- Hagyományos. A megszokott dobozolós technológiát követi.

A teljesség kedvéért említsünk meg néhány példát a használatára.

- A DNS nagyon jó példa, ugyanis az vegyesen használ UDP-t, illetve TCP-t. Ha elfér az üzenet egy UDP csomagban - tipikusan névfeloldási kérés, illetve válasz - akkor az UDP lesz a befutó. Ha több csomagra lesz szükség - tipikusan zónatranszfer - akkor viszont a TCP.

- Van, amikor maga az alkalmazás szintű protokoll foglalkozik a csomagok szállításának megbízhatóságával. Ilyen pl. a TFTP.

- Van, amikor a feladó periodikusan küldi el az információkat. Ilyenkor nem gond, ha az egyik küldemény nem érkezik meg, hiszen hamarosan megy a másik. Ezt tudja a korábban már említett RIP.

- Végül a multicast. A TCP csak egy-egy tipusú kapcsolatot tud kezelni.