

Ellenőrző kérdések a ZH témaköréből

Hálózati architektúra

1. Analógia az emberi kommunikáció és a gépi kommunikáció között.

Emberi kommunikáció:

- Közvetlen kommunikáció mondatokkal: A két ember A és B. A mondatokat generál, amire B válaszol(hat).
- Közvetett kommunikáció, ha távol vannak egymástól: A kommunikáció során megkülönböztetjük a valós utat és a virtuális utat. A virtuális úton A és B beszélgetése továbbra is úgy folyik, mintha egymás mellett lennének. A kommunikáció ténylegesen a valós úton keresztül zajlik. Ehhez a felhasználók egy távközlési szolgáltató (pl.: telefonhálózat) szolgáltatásait veszik igénybe. A szolgáltatóhoz a SAP-on (Service Access Point, szolgáltatás-elérési pont) keresztül csatlakoznak.

Gépi kommunikáció:

- Magában álló rendszer esetén: Adott egyetlen rendszer és azon az OS segítségével kommunikálnak egymással a folyamatok. IPC (Inter Process Communication)
- Egymással kommunikáló gépek esetén: A cél, hogy az A gépen futó X alkalmazási folyamat és a B gépen futó Y alkalmazási folyamat úgy tudjon kommunikálni, mintha a folyamatok ugyanazon a gépen futnának, vagyis a kommunikáció részleteiről ne kelljen tudnia egyik folyamatnak sem. Ennek érdekében A-t és B-t el kell látni egy, ezt lehetővé tévő kommunikációs hardver-szoftver együttesel (C folyamat). Ennek az egységnek a szolgáltatásait veszik igénybe a gépeken futó alkalmazási folyamatok. Ha X-nek kellene Y adatai, X megkéri C-t, hogy hozzon létre kapcsolatot B-vel, annak érdekében, hogy kommunikálhasson Y-nal. X-nek és Y-nak nem kell tudni a részletekről.

Analógia:

Közvetlen kommunikáció mondatokkal ~ Magában álló rendszer

Közvetett kommunikáció ~ Egymással kommunikáló gépek

2. Hogyan valósul meg a kommunikáció a távoli felhasználók között?

Primitívekkel valósul meg:

- **request** - req (kérés): a kommunikációt kezdeményező fél küldi a szolgáltatónak, hogy kezdeményezze a kapcsolat létesítését. (X a kagylót felveszi, tárcsáz)
- **indication** - ind (jelzés): a szolgáltató jelzi az elérni kívánt fél részére, hogy szeretnének vele kapcsolatot teremteni. (Y telefonja csörög)
- **response** - res (válasz): az elérni kívánt fél jelzi a szolgáltató felé, hogy kész a kapcsolatfelvételre. (Y felveszi a telefont)
- **confirmation** - conf (megerősítés): a szolgáltató értesíti a kezdeményező felet, hogy a távoli fél elfogadja a kapcsolatfelvételt, a kapcsolat létrejött. (X hallja, hogy a kicsöngés abbamarad, beszélhet)

A felhasználók közötti kommunikáció virtuális és nem közvetlen. Ahhoz, hogy az alkalmazások kommunikálni tudjanak egymással, egy egységes felületre és egy kommunikációs alrendszerre van szükség, mely biztosítja a két kommunikáló fél között a kapcsolat kiépülését. A tényleges kommunikáció az adatátviteli hálózaton keresztül zajlik.

3. Ismertesse a szolgáltató-felhasználó modellt és elemeit!

A szolgáltató-felhasználó modellben a két egyenrangú felhasználó kommunikál egy szolgáltatón keresztül (ez lehet többretegű, ahol az rétegek az alacsonyabb szintű réteg szolgáltatásait veszik igénybe, azt kiegészítve valamivel, így *értéknövelt* szolgáltatást nyújtva). A szolgáltató által nyújtott szolgáltatásokat a felhasználó a SAP-on (szolgáltatás elérési pont) keresztül tudja igénybe venni, ehhez azonban ismernie kell az interfészét. A hangsúly a hálózati szolgáltató és a felhasználó(k) közötti együttműködésen van. Az együttműködés célja olyan hálózati szolgáltatás megvalósítása, melynek segítségével a felhasználók között adattovábbítás végezhető.

Elemei: felhasználók, és az (esetleg többretegű) szolgáltató (A szolgáltatást több *funkcionális elem* biztosítja, lásd később).

4. Mi a szolgálat definíciója? Mit kell tudni a felhasználónak a szolgálatról?

- **Szolgáltató szempontjából:** Azon tevékenységek, műveletek, függvények, eljárások halmaza, amelyet valaki vagy valamilyen szervezet **nyújt másvalakiknek**, azok munkájának vagy életének megkönnyítésére vagy gazdaságossá tételére.
- Felhasználó szempontjából: Azon mások által végzett tevékenységek, műveletek, függvények, eljárások halmaza, amelyet - mint felhasználó - **igénybe veszünk** annak érdekében, hogy munkánkat vagy életünket megkönnyítsük vagy gazdaságossá tegyük.

A felhasználónak nem kell ismernie a szolgáltató belső működését, a szolgáltatóhoz **szabványos interfészen** csatlakozik (csupán ezt kell ismerni), lehet összeköttetés alapú (telefon) vagy összeköttetés nélküli (posta)

5. Mit nevezünk számítógép-hálózati architektúrának?

Olyan architektúra, amelynek konkrét célja a távoli számítógépeken futó alkalmazási folyamatok között magasszintű adatcsere megvalósítása. lásd még 8. pont.

Réteg:

- egy réteg valamilyen szolgáltatásnyújtás érdekét szolgálja
- a réteg egy funkció- vagy feladathalmaz adott szintű szolgálat nyújtására
- elemei a funkcionális elemek (FE)
- a szolgáltatás egy adott szintjét valósítják meg
- a köztük lévő relációkat protokollok határozzák meg

6. Mit nevezünk hivatkozási modellnek? Milyen kapcsolata van a hálózati architektúrával?

Többrétegű, hierarchikus beágyazott modell, a számítógép architektúráját írja le. (+ ábra valahonnan) pl.: ISO-OSI, TCP/IP, (SNA)

7. Mi az előnye a hivatkozási modellnek?

Szabványosítás menetének megkönnyítése, követelmények megadásának megkönnyítése, áttekinthetőség, fejleszthetőség, dokumentálhatóság, struktúráltság.

- Bonyolult rendszer leírása, átláthatóvá, megfoghatóvá tétele
- Szabványosítás megkönnyítése, ugyanis az egyes rétegekkel szembeni követelmények könnyebben leírhatók
- Követelmények megfogalmazása, leírása könnyebbé válik
- Az egyes rétegek viszonyát és a rendszer felépítését jól lehet ezen keresztül kezelni
- A rendszer felépítését jól tükrözi, mert koncepcionális (elvi megoldásokat ad, míg a konkrét szoftver termékek esetleg nem minden elemét tartalmazzák)
- Dokumentálhatóság megkönnyítése, mivel a rendszer struktúrált és áttekinthető
- Irányelvet ad arra, hogyan célszerű a témakört oktatni: alulról felfelé építkezve, egy-egy szint megismerése után továbblépve, az addigi színteket adottnak tekintve

8. Ismertesse az architektúra fogalmát általában és konkrétan a számítógép-hálózati architektúráét, annak elemeit és relációit (réteg, SAP, protokoll, interfész, primitív, PDU, SDU, PCI, beágyazás)!

- **Architektúra**

Valamilyen cél érdekében alkalmazandó funkcionális elemek (blokkok, modulok) halmaza, és a halmaz elemei közötti relációk összessége. Mind a halmaz elemeit, mind a köztük lévő relációkat technológiai és esztétikai szempontok befolyásolhatják. Számítógéphálózati architektúra (5. pont).

Ennek elemei:

- **Rétegek**, ahol egy réteg valamilyen szolgáltatásnyújtás érdekét szolgálja. A kommunikációs rendszer egymástól jól elkülönülő, független részei, valamely szabványos interfésszel kapcsolódnak egymáshoz.
- **SAP**: Szolgáltatáslézési pont, Service Access Point.
- **Protokoll** szitaktikai, szemantikai és logikai szabályok összessége.
- **Interfész** rétegek közötti kapcsolat megvalósítása - ?? milyen kapcsolatban van az SAP-vel?:
- **Primitív** a szolgáltató legfelső rétege kommunikál a felhasználókkal, fajtáit a 2. pontban.
- **PDU** Protokoll adategység, Protocol Data Unit (=PCI+SDU).
- **SDU** szolgáltató adategység (Service Data Unit)
- **PCI** Protokoll vezérlő információ, Protocol Control Information.
- **Beágyazás** Az N. réteg funkcionális elemei a fölöttük lévő rétegtől kapott adatot szolgáltató tárgyának tekintik.

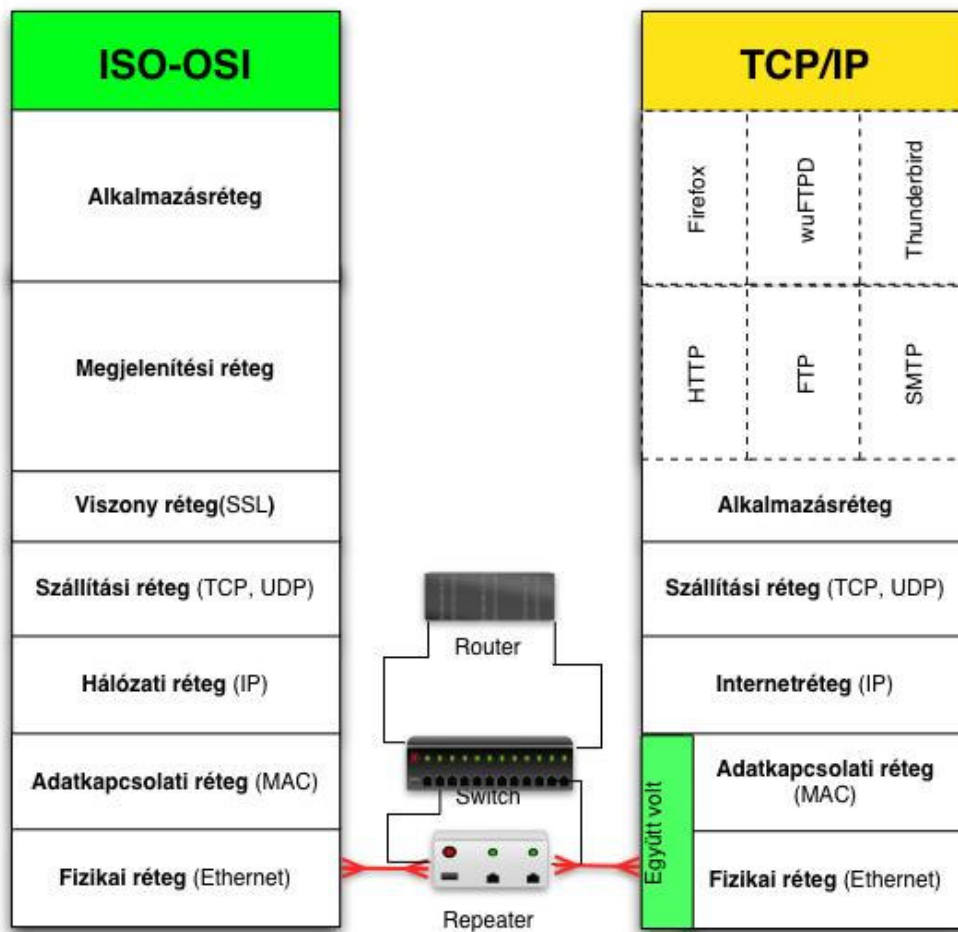
9. Adatszerkezetek segítségével mutassa be a beágyazódási folyamatot!

Az N. réteg funkcionális elemei a fölöttük lévő rétegtől, (N+1). rétegtől kapott adatot szolgálatuk tárgyának tekintik, azaz az (N+1). PDU az N. réteg SDU-ja lesz, ezt az SDU-t megváltoztatni nem szabad. Tehát a kapott adatot az N. rétegnek változatlanul kell továbbadnia. Továbbadás előtt azonban kiegészíti a saját maga által generált fejléccel (N. PCI: az N. FE saját belső vezérlő információja (sorszám, cím, útvonal, parancs, válasz, stb.)). A fejléccel ellátott N. SDU-ból az N. réteg saját N. PDU-ja lesz. Ezt a PDU-t az N. réteg továbbadja az (N-1). rétegnek az (N-1). interfészen keresztül.
A beágyazódás, ill. fejrésszel való ellátás addig tart, amíg el nem jut a fizikai rétegig (bitek szintjéig) és ott struktúrátlan biztsorozatok formájában küldi el a vevő oldal legalsó szintjének.

10. Nyílt rendszer definíciója. Nyílt rendszerek összekapcsolása.

Előre meghatározott kommunikációs követelményeknek eleget tevő rendszer. Több ilyen rendszer összekapcsolásával is nyílt rendszer jön létre, és így ezekhez más, ugyanilyen rendszerek is kapcsolódhatnak (szabványok, ajánlások figyelembevételével lehet egy ilyen rendszert előállítani és ilyenhez kapcsolódni).

11. Az OSI 7 rétegű hivatkozási modell ismertetése.



Felső rétegek: alkalmazási, megjelenítési és viszony réteg. Hálózatifüggetlen funkciókat alkalmaznak, alkalmazásorientált feladatok vannak.

- Alkalmazási réteg: elosztott, adattovábbító szolgálat.
- Megjelenítési réteg: az alkalmazási folyamat által átvinni kívánt adat formátumát, megjelenítési formáját megváltoztathatja (pl. titkosítás)
- Viszony réteg: felhasználói adatok átvitelének szervezésével foglalkozik. Feladata, hogy a felhasználó ne vegye észre, ha a hálózat belső működésével probléma van.

Alsó rétegek: szállítási, hálózati, adatkapcsolati és fizikai réteg. Hálózatifüggő funkciókat tartalmaznak.

- Szállítási réteg: feladata a hosztok közötti adattovábbítás, a hálózaton történő megbízható adatszállítás.
- Hálózati réteg: feladata a forrás-csomópont és nyelő-csomópont közötti adattovábbítás a kommunikációs alhálózaton.
- Adatkapcsolati réteg: feladata a csomópontok közötti adattovábbítás.
- Fizikai réteg: feladata, hogy a struktúrátlan bitsorozat fizikailag is átmenjen egyik csomóponttól a másikra.

Az egyes rétegek között a nevüknek megfelelő protokollok teremtenek kapcsolatot.

12. Két alkalmazási folyamat kommunikációja az OSI modell alapján.

A legfelső réteg (7.) kommunikációja:

- a 7. rétegben futó folyamat előállít egy m üzenetet
- a 6/7 interfész által a 7.-ből a 6. rétegbe kerül
- különböző átalakítások (tördelés, fejrész hozzáadás stb.) után eljut a legalsó rétegbe
- a vevő oldali gépen az üzenet rétegről rétegre halad felfelé, miközben folyamatosan megszabadul a fejrészekről

protokollok közötti kommunikáció:

- vízszintes irányú (virtuális)
- az adó n. protokollja kommunikál a vevő n. protokolljával
- a tényleges kommunikáció függőleges irányú

13. Hogyan kommunikál két társentitás?

Azokat a funkcionális egységeket, amelyek a különböző gépeken az egymásnak megfelelő rétegeket magukba foglalják, társfolyamatoknak nevezzük. Ezek a társfolyamatok kommunikálnak egymással a protokollok felhasználásával. A valóságban nem az egyik gépen lévő n. réteg kommunikál a másik gépen lévő n. réteggel. Minden egyes réteg adat- és vezérlőinformációkat ad át az alatta elhelyezkedő rétegnek, egészen a legalsóig: a fizikai rétegen zajlik a tényleges kommunikáció.

14. Összeköttetés alapú és összeköttetés nélküli szolgálatok jellemzői, azonosságok, különbségek.

- **Összeköttetés alapú** A szolgálatot igénybevevő felhasználó létrehozza az összeköttetést, használja, majd lebontja. A bitsorrend általában megmarad pl: telefonrendszer
- **Összeköttetés nélküli** Minden üzenet a többlettől független útvonalon továbbítódik pl: posta

15. Mik a szolgálat primitívek osztályai (típusai), hogyan működnek?

A távoli felhasználók közötti kapcsolat kialakításában a szolgáltató játszik nagy szerepet, feladata a felhasználók által igényelt szolgáltatás teljesítése. Ezen szolgáltatások megvalósulása érdekében szükség van a szolgáltató és az ügyfél közötti kommunikációra, ez a kommunikáció szabványos primitívekből áll.

fajtái: lsd 2. pont

16. Mutasson egy példát a megerősített szolgálatra (pl. telefon)!

A központ veszi a jelzést, feldolgozza, **értesíti** a másik felhasználót (pl. *kicseng* a telefon), ha **válaszol** (pl. *felemeli* a kagylót), akkor **visszaigazolást** küld (response), ez a visszaigazolás a hívónál egy konfirmációt eredményez (pl. *meghallja* a hívott fél hangját). A felek között van response és konfirmáció, ezért ez egy megerősített szolgáltatás

17. A szolgálatok és a protokollok közötti kapcsolat.

A szolgálat nem más, mint olyan primitívek halmaza, amelyet egy adott réteg a felette levő rétegek számára biztosít. A szolgálat azt definiálja hogy egy réteg a felhasználó nevében milyen műveletet képes végrehajtani, de annak implementációjáról nem mond semmit.

A protokoll ezzel szemben olyan szabályok halmaza, amelyek azt modják meg, hogy milyen legyen a formátuma, és mi legyen a jelentése azoknak a kereteknek, csomagoknak, üzeneteknek, amelyeket az adott réteg társentítései küldözgetnek egymásnak.

Azaz: a szolgálatok a rétegek közötti interfésszel kapcsolatosak, míg a protokollok a különböző gépeken elhelyezkedő társentítések között elküldött csomagokkal kapcsolatosak.

(analógia a programozási nyelvekkel: szolgálat=absztrakt objektum, protokoll= a szolgálat implementációja)

18. Írja le a hivatkozási modell alapján két távoli funkcionális elem között a kapcsolatfelvétel, adatátvitel és kapcsolatbontás folyamatát a szállítási szolgálat igénybevételével!

Összeköttetés-alapú szolgálat (connection-oriented service) esetén 3 fázisról beszélhetünk:

1. összeköttetés létesítése
2. adatátvitel
3. összeköttetés bontása.

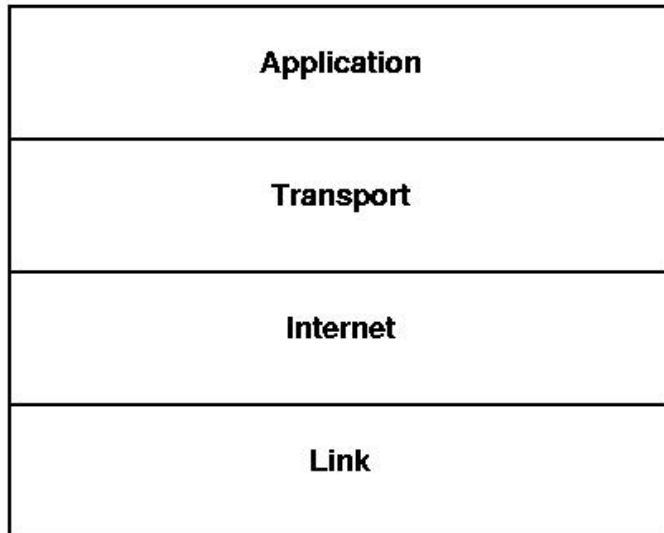
Az egyes fázisokban használt primitívek felépítése általános esetben $X_Y.z(\text{paraméterek})$, ahol:

* X = réteg azonosítója

- - viszony (session, S)
 - szállítási (transport, T) * Y = a szolgálat megfelelő fázisa
 - összeköttetés létesítése (connect, CON)
 - adatátvitel (DATA)
 - összeköttetés bontása (disconnect, DISC) * z = a primitív nevének rövidítése
 - kérés (request, req)
 - bejelentés (indication, ind)
 - válasz (response, resp)
 - megerősítés (confirmation, conf)

19. Az eredeti TCP/IP hivatkozási modell ismertetése. Eltérések az OSI modellhez képest.

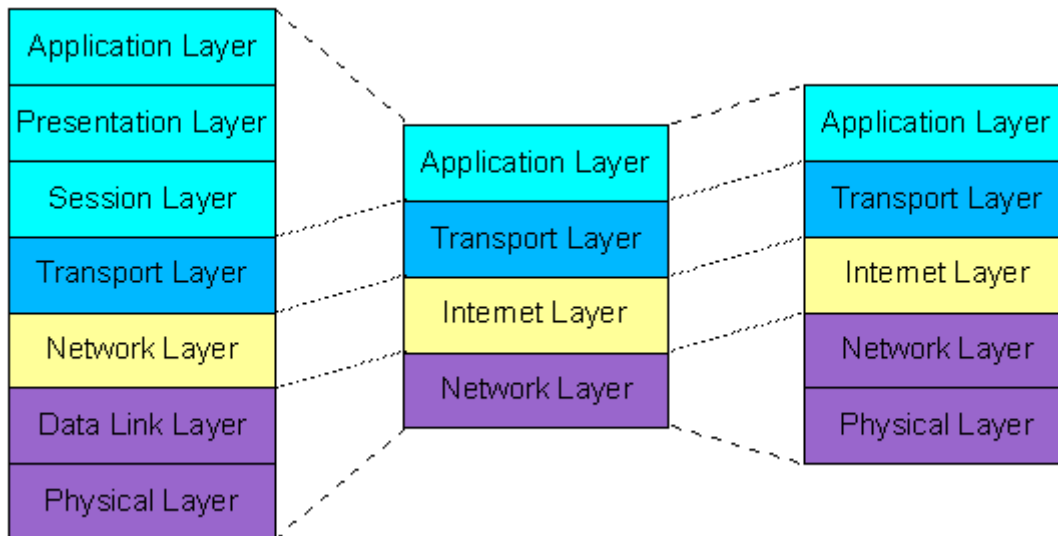
TCP/IP Model



- Alkalmazási protokoll (pl FTP, SNMP, HTTP)
- TCP protokoll (pl TCP, UDP)
- IP protokoll: IP
- Network interface protokoll (pl PPP)

("többet nem tudok mondani, kérem kapcsolja ki")

20. A hibrid (javított) TCP/IP architektúra jellegzetességei.



- Alkalmazási réteg (Application/Process): FTP,SMTP,SNMP,HTTP
- TCP (Transmission Control Protocol) réteg (Host-Host): TCP,UDP
- IP réteg: Internet (IP)

Network Access réteg (a köv. kettő együtt)

- Adatkapcsolati réteg (Datalink): PPP
- Fizikai réteg (Physical): bitfolyam

Előnyök:

- kevesebb réteg -> gyorsabb , kevesebb erőforrás kell

Hátrányok:

- nincs megjelenítési és viszony réteg -> külön kell foglalkozni az alkalmazási rétegben pl a titkosítással, ami a 7 rétegű modellben a megjelenítési réteg feladata

-- [GorcsGergely](#) - 2005.10.24.
-- [Dani](#) - 2005.10.25.
-- [Zoz](#) - 2005.10.26.
-- [adamo](#) - 2005.10.27.
-- [SzaMa](#) - 2005.10.27.
-- [Aadaam](#) - 2005.10.27.
-- [KornaiTamas](#) - 2005.10.27

Revision: r1.10 - 2005.10.27 - 23:42 - [KornaiTamas](#)
[Infoalap](#) > [SzgHalok](#) > SzgHaloZhArch