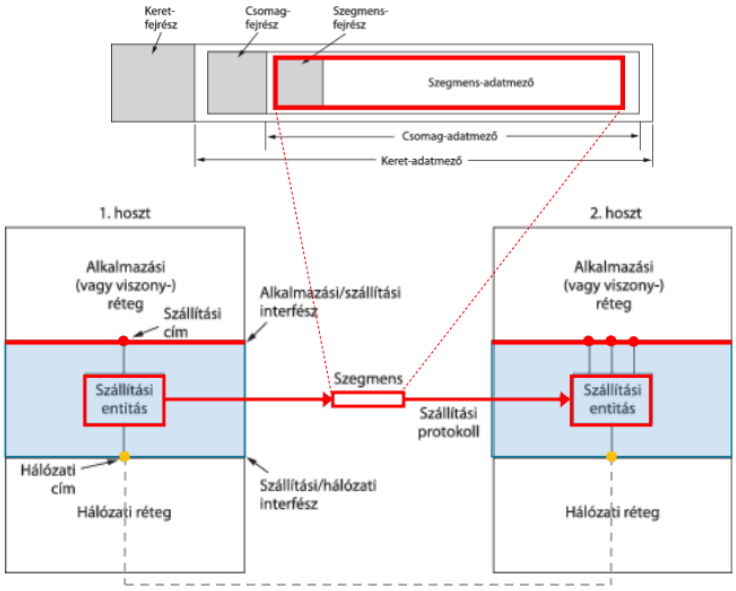
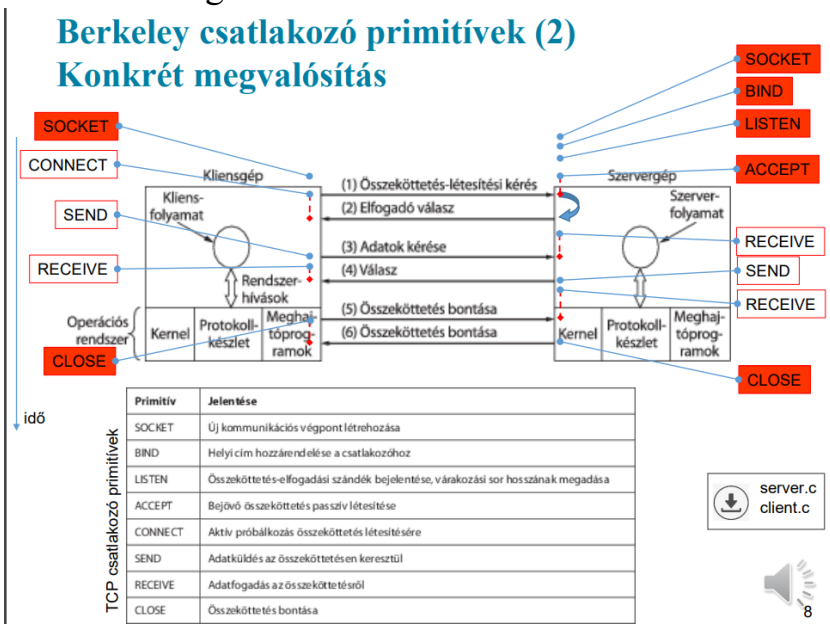
1. Mi a szállítási réteg feladata? Milyen szolgáltatásokat nyújt a hálózati réteg?
    a. A hálózati réteg:
        i. Két végpont közötti csomagtovábbítás
            - Datagram
            - Virtuális áramkör
        ii. Nem megbízható
        iii. Nagyrészt útválasztókon fut
            - Hiba esetén a felhasználó nem tud beavatkozni
    b. Szállítási réteg:
        i. Forrásgép egy folyamatától a célgép egy folyamatáig
        ii. Megbízható szolgáltatás
        iii. Használja a hálózati réteg szolgáltatásait
        iv. A felhasználó gépén fut
            - Teljes felhasználói kontroll
            - Társentitások tudnak egyeztetni
                a. Pl.: megérkezett? Ha nem, újraküldés.
            - Így megbízhatóbb tud lenni, mint az alatta lévő réteg
2. Rajzolja fel a hálózaton közlekedő adatok beágyazását (fejrészekkel). Nevezze meg az egyes elemeket.



3. Ismertesse a Berkeley csatlakozó primitíveket. Hogyan használjuk az egyes rendszerhívásokat? Milyen üzenetek közlekednek a hálózaton a rendszerhívások meghívásakor?



Primitív	Jelentése
SOCKET	Új kommunikációs végpont létrehozása
BIND	Helyi cím hozzárendelése a csatlakozóhoz
LISTEN	Összeköttetés-elfogadási szándék bejelentése, várakozási sor hosszának megadása
ACCEPT	Bejövő összeköttetés passzív létesítése
CONNECT	Aktív próbálkozás összeköttetés létesítésére
SEND	Adatküldés az összeköttetésen keresztül
RECEIVE	Adatfogadás az összeköttetésről
CLOSE	Összeköttetés bontása

4. Mire való a SOCKET primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Új kommunikációs végpontot hoz létre
    b. Mindkettő fél használja
5. Mire való a BIND primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Helyi címet rendel hozzá a csatlakozóhoz
    b. Szerver használja
6. Mire való a LISTEN primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Összeköttetés-elfogadási szándék bejelentése, várakozási sor hosszának megadása
    b. Szerver használja
7. Mire való az ACCEPT primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Bejövő összeköttetés létesítésére szolgál
    b. Szerver használja
8. Mire való a CONNECT primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Aktív próbálkozás összeköttetés létesítésére
    b. Kliens használja

9. Mire való a SEND primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Adatküldés az összeköttetésen keresztül
    b. Mindkettő használja
10. Mire való a RECEIVE primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Adatfogadás az összeköttetésen keresztül
    b. Mindkettő használja
11. Mire való a CLOSE primitív? Melyik fél használja (szerver/kliens/mindkettő)?
    a. Összeköttetés bontása
    b. Mindkettő használja
12. Mit jelent a TSAP és az NSAP? Hogy nevezzük ezeket a csatlakozókat az internet világában?
    a. TSAP (Transport Service Access Point)
        i. Jelentése: A szállítási (transport) réteg végpontjait azonosítja.
        ii. Feladata: Egy adott alkalmazás számára biztosítja a kommunikációt a szállítási rétegben.
        iii. Példa az internet világában: A portszám.
            - Az IPv4 és IPv6 protokolloknál a portok határozzák meg, hogy melyik alkalmazás vagy szolgáltatás kapja meg az adott szállítási réteg (pl. TCP vagy UDP) által továbbított adatokat.
            - Például:
                a. HTTP szolgáltatás portja: 80
                b. HTTPS szolgáltatás portja: 443
    b. NSAP (Network Service Access Point)
        i. Jelentése: A hálózati réteg végpontjait azonosítja.
        ii. Feladata: Egy adott hálózati interfész számára azonosítja a hálózati rétegben használt címeket.
        iii. Példa az internet világában: Az IP-cím.
            - Az IP-cím (IPv4 vagy IPv6) azonosítja az eszközt a hálózaton belül, és lehetővé teszi, hogy az adatokat a megfelelő helyre továbbítsák.

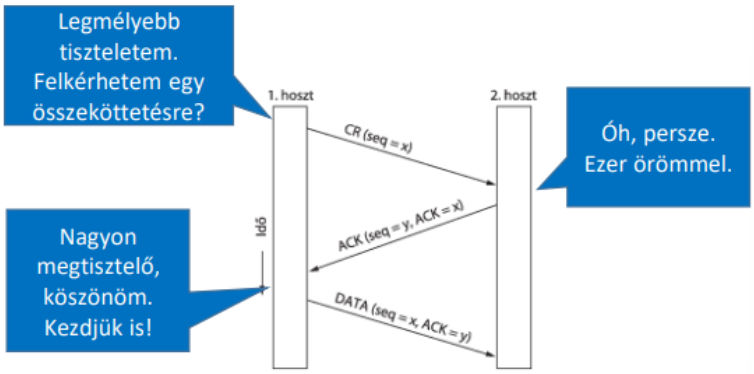
13. Honnan tudhatja egy kliens gép a szolgáltatás TSAP címét?
    a. Állandó cím, amit egy adatbázis tárol
    b. Sok szolgáltatás **fix portszámot** használ, amelyet az operációs rendszer vagy egy adatbázis tárol, így a kliens előre tudhatja, hogy melyik porton találja a szolgáltatást
        i. Pl.: levelezés – 25
        ii. Pl.: Unix: /etc/services
    c. Portszolgáltató (portmapper)
        i. Speciális folyamat
        ii. Címe ismert/állandó
            - a portmapper folyamat egy **fix portszámon** fut
        iii. dinamikusan kezeli a szolgáltatások portszámait
        iv. feladata, hogy a kliensek számára megmondja, melyik porton érhető el egy adott szolgáltatás
        v.
14. Ismertesse a portszolgáltató működését.
    a. ÖK (összeköttetés) létesítése portszolgáltatóval (PSZ)
    b. Kérés küldése: hol van a szolgáltatás?
    c. PSZ visszaküldi a szolgáltatás portszámát
    d. ÖK bontása PSZ-val
    e. ÖK létesítése a szolgáltatással…
15. Ismertesse a kezdeti összeköttetés protokoll működését. Hogyan működik a folyamatszerver (folyamatszolgáltató)?
    a. kezdeti összeköttetés protokoll működése:
        i. A szerver folyamatot indít egy **ismert portszámon.** A szerver a **passzív állapotban lévő socketjével (foglalatával)** figyel egy adott hálózati interfészen és porton
        ii. A kliens létrehoz egy **aktív socketet**, amelyet egy ideiglenes helyi porthoz köt. Ez lesz a kliens oldali kapcsolati végpont. A kliens az ismert **szerver címet** (IP-cím és port) használva kezdeményez egy kapcsolatot
        iii. A szerver elfogadja a kliens kezdeményezését, és létrehoz egy **új socketet**, amely a konkrét klienssel történő kapcsolatot kezeli. Ez az új socket már a kliens és a szerver közötti adatátvitelre szolgál, míg az eredeti socket a további kapcsolatkérések figyelését végzi.
        iv. A kliens és a szerver a felépített kapcsolaton keresztül adatokat cserélhet
        v. A kapcsolat lezárását bármelyik fél kezdeményezheti, amelyet a TCP-protokoll megfelelő mechanizmusai hajtanak végre.

b. Folyamatszerver (folyamatszolgáltató)
    i. Több portot is figyel
    ii. Ha kérés érkezik, akkor
        - elindítja a megfelelő folyamatot
        - átadja neki az ÖK-t
    iii. Folyamatszerver működése:
        - Az 1. hoszt folyamata levelet akar küldeni a 2. hoszt 25-ös portján keresztül
        - A 2. hoszton nem fut a levelező szerver, csak a folyamatszerver (FSZ)
        - A kapcsolatot a FSZ építi ki a 25-ös porton
        - A FSZ elindítja a levelező szervert és átadja neki a kapcsolatot
        - FSZ tovább figyel a többi porton

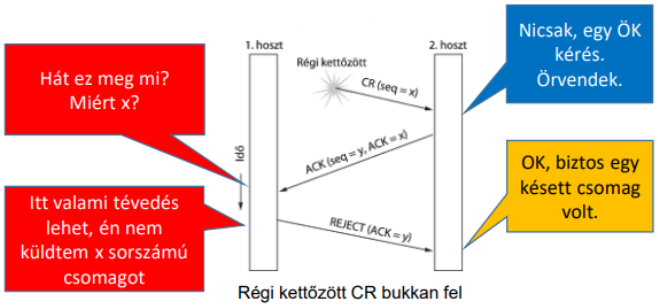
16. Mit jelent a „jól ismert port”? Soroljon fel legalább 3 jól ismert portot.
    a. Szerverek gyakran „ismert” portokon vannak
    b. Előre definiálva vannak bizonyos szolgáltatások számára
    c. 1024 alatti (0-1023) portok tartoznak ide általában
    d. Pl.:
        i. SSH (TCP, UDP): 22
        ii. FTP (TCP): 20, 21
        iii. SMTP (TCP): 25
        iv. HTTP (TCP): 80
        v. HTTPS (TCP): 443
17. Hogyan kezeljük a duplikált csomagok problémáját? Milyen szabályt kell betartanunk a sorszámok ismétlődésével kapcsolatban?
    a. Sorszámozzuk a csomagokat
        i. A sorszámok egy tartományban mozognak, utána átfordulnak
            - Pl. 32 bites egész számok
        ii. Ha az adási sebesség korlátos, akkor egy T ideig biztosan nem ismétlődhetnek a sorszámok
            - Ha mégis, akkor az egy kóbor/késő csomag, amit már megismételtünk
18. Hogyan gondoskodunk a sokat késő csomagok eliminálásáról?
    a. De mi van a nagyon sokat késő csomagokkal (>T)?
        i. Ezeket ki kell irtani
        ii. Módszerek:
            - Ugrásszámláló alkalmazása (ha túl sok ugráson át bolyong, akkor eldobjuk)

- Időcímke alkalmazása (ha túl hosszú ideig bolyong, akkor eldobjuk)
    a. Bonyolult (szinkronizált órák kellenek)
    b. Helyette az ugrásszámlálót használjuk

19. Ismertesse a háromutas kézfogásos összeköttetés-létesítés működését.
    a. CR (Connection Request)
        i. Connection request
        ii. Saját sorszám: x
    b. ACK (Acknowledge)
        i. Saját sorszám: y
        ii. Nyugtázza x-et
    c. DATA
        i. Nyugtázza y-t
    d. A gyakorlatban
        i. a kezdeti sorszám (x, y) álvéletlen szám
        ii. Biztonsági okból:
            - Ne legyen megjósolható, így támadható

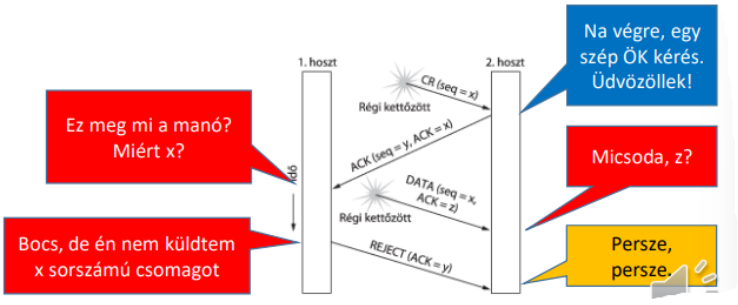


20. Ismertesse a háromutas kézfogásos összeköttetés-létesítés működését, amennyiben egy régi kettőzött CR üzenet bukkan fel. Hogyan kezeli ezt a hibát a protokoll?



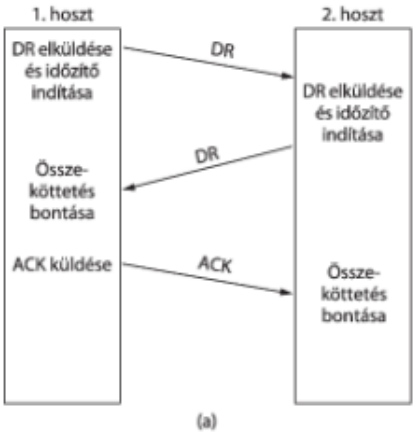
Régi kettőzött CR bukkan fel

21. Ismertesse a háromutas kézfogásos összeköttetés-létesítés működését, amennyiben egy régi kettőzött CR üzenet és egy régi kettőzött ACK üzenet bukkan fel. Hogyan kezeli ezt a hibát a protokoll?

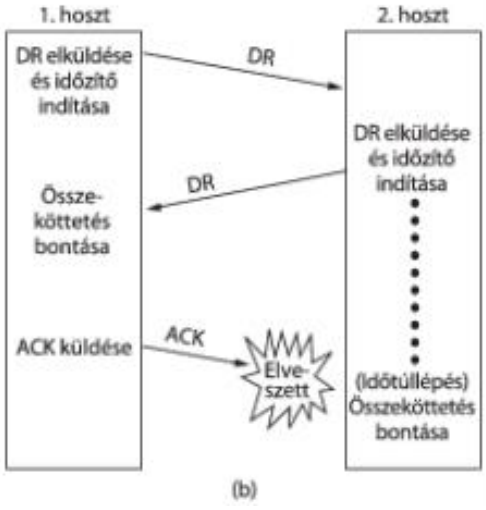


Kettőzött CR és kettőzött ACK esete

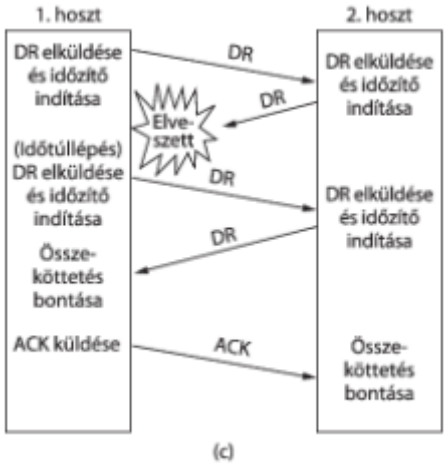
22. Ismertesse a szimmetrikus és aszimmetrikus bontás működését. Melyik megközelítés előnyösebb és miért?
    a. Aszimmetrikus bontás
        i. Mint telefonálás esetén
        ii. Az egyik fél bont és az ÖK megszakad
        iii. Ez adatvesztéssel járhat
    b. Szimmetrikus bontás
        i. Mindkét irányt külön kezeljük
        ii. Mindkét irányt függetlenül bontjuk le
        iii. Ha az egyik irányt lebontottuk, a másikban még küldhetünk adatot
    c. Nem lehetünk benne biztosak, hogy a másik fél is bontani akarja-e az ÖK-t. Ezért nincs tökéletesen működő protokoll.
23. Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldást. Hol alkalmazunk időzítőket a protokollban és miért?
    a. Háromutas kézfogás + időzítő
    b. DR: DISCONNECT REQUEST
        i. Egyirányú bontás jelzése
    c. Normál működés
    d. Elveszett ACK.
        i. 2. hoszt időtúllépéssel bont
    e. Elveszett válasz DR
        i. 1. hoszt ismételt DR-t küld
    f. Elveszett válasz és minden további üzenet
        i. Mindkét hoszt időtúllépéssel bont



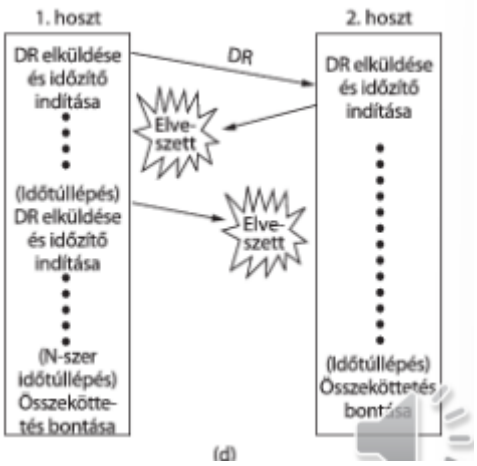
24. Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben az ACK üzenet elveszik.
    a. 2. hoszt időtúllépéssel bont



25. Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben a válasz DR üzenet elveszik.
    a. 1. hoszt ismételt DR-t küld



26. Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben a válasz DR üzenet és minden további üzenet elveszik.
    a. Mindkét hoszt időtúllépéssel bont



27. Miért szükséges a hibakezelés a szállítási rétegben?
    a. Hibakezelés:
        i. Adatok megfelelő biztonsággal (hibátlanul) érkezzenek meg
    b. Az adatkapcsolati rétegben ugrásonként ellenőrzünk. De mi történik, ha pl. egy router belsejében sérül az adat?
    c. Mindenképpen szükséges a végponttól végpontig ellenőrzés
    d. Nagy sávszélesség-késleltetés szorzat ◊ nagy pufferméret. Ezt okosan kell kezelni
    e. Mindig van újraküldés, kijavítja az összes, esetleg még előforduló hibát

28. Mi a forgalomszabályozás feladata?
    a. Az adó és vevő sebességét szinkronizálja
    b. Egy gyors adó ne töltsön túl egy lassú vevőt
    c. Kezelés:
        i. Vevő információt küld az állapotáról (ablakméret)
        ii. Adó ennek megfelelően ad (elküldött csomagok száma)
        iii. Csúszóablakos protokoll
29. Ismertesse a csúszóablakos protokoll működését.
    a. változó méretű ablakokat (pufferméret) használ
    b. külön választja a nyugtázást és a pufferkezelést
        - Egyszerű példa:
            - A ad
            - B vesz
            - Sorszám 4 bites (0-15)
        - B minden üzenete tartalmaz:
            - nyugta (sorszámmal)
            - rendelkezésre álló pufferméret (vevő oldali ablakmérettel)
        - Valóságban szimmetrikus!
            - A is nyugtázza B üzeneteit
            - A is küld ablakméretet B-nek

30. A csúszóablakos protokollban a következő üzeneteket látjuk:
    A->B sorszám=7, adat=d7
    A<-B nyugta=4, puffer=6
    Hány üzenetet küldhet ezután az A állomás, ha nem kap B-től további üzenetet?
    a. nyugta=4 azt jelenti, hogy a B állomás az eddig sikeresen átvett üzenetek sorszámának a 4-es üzenetig terjedő részét visszaigazolta
    b. puffer mérete (6) azt jelenti, hogy az A állomás a sorszám 5-től kezdve a sorszám 10-ig terjedő üzeneteket küldhet (mivel a puffer ablak mérete 6)
    c. Az A állomás a sorszám 5-től kezdve folyamatosan küldhet üzeneteket, mivel a B az összes, 4-es sorszámú üzenetet visszaigazolta, és az ablak puffer mérete 6
    d. Ezért a következő üzenetek sorszáma, amelyeket az A állomás küldhet, a 5, 6, 7, 8, 9, 10 (összesen 6 üzenet) lesz, amíg új nyugtát nem kap a B-től

31. Kis idő múlva a következő két üzenet látjuk a hálózaton:

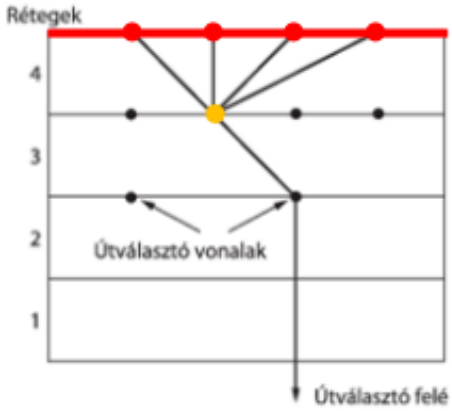
    A<-B nyugta=12, puffer=3

    A->B sorszám=13, adat=d13

    Mely sorszámú üzeneteket küldheti még el az A állomás (amíg nem kap B-től újabb üzenetet)?

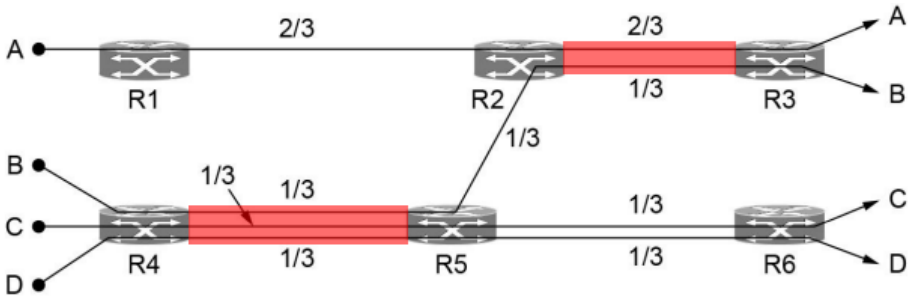
    a. Nyugta sorszáma (12) azt jelenti, hogy a B állomás visszaigazolta az összes üzenetet az 1-től 12-ig terjedő sorszámokig, tehát az A állomás biztosan tudja, hogy ezek az üzenetek megérkeztek
    b. A puffer mérete (3) azt jelenti, hogy az A állomás maximum 3 üzenetet küldhet el anélkül, hogy újabb nyugtát kapna
    c. Az A állomás a 13-as sorszámú üzenetet már elküldte, így az ablakban most 2 hely van a puffer méretének köszönhetően. Ezért az A állomás csak 2 üzenetet (14-es és 15-ös sorszámúakat) küldhet el a 13-as üzenet után, mielőtt újabb nyugtát kapna.

32. Mit jelent a nyalábolás a szállítási rétegben?
    a. Több szállítási összeköttetés ugyanazt a hálózati címet használja
    b. Több TSAP → egy NSAP
    c. Ez a tipikus, hiszen általában 1 hálózati cím tartozik egy géphez
    d. TCP ezt használja



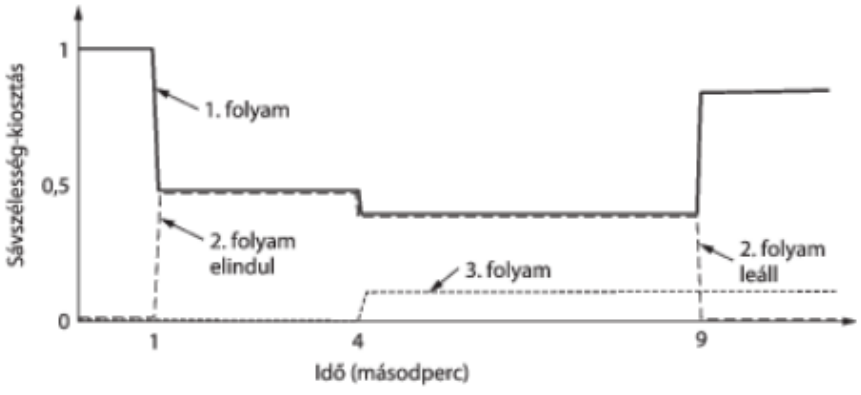
33. Mit jelent a fordított nyalábolás a szállítási rétegben?
    a. Egy szállítási összeköttetés több hálózati címet (interfészt) is használ
    b. Egy TSAP → több NSAP
    c. Ok: sávszélesség megnő
    d. k db. hálózatai összeköttetés: k-szoros sávszélesség
    e. SCTP (Stream Control Transmission Protocol) is ilyen megoldást használ

34. Mi a torlódáskezelés feladata?
    a. Hatékonyság biztosítása
        i. Felajánlott forgalom (offered load)
        ii. Hasznos átbocsátóképesség (goodput)
            - Egy darabig a felajánlott forgalommal együtt nő, majd:
            - Torlódás:
                a. Goodput lassul, majd visszaesik
                b. Késleltetés nő
        iii. Legjobb teljesítőképesség:
            - A sávszélességet a késleltetés növekedéséig foglaljuk le
    b. Sávszélesség igazságos elosztása
        i. Maximum-minimum igazságosság
            - Egy kiosztás maximum-minimum igazságos, ha egy folyamnak kiosztott sávszélesség nem növelhető anélkül, hogy egy másik, kisebb vagy azonos kiosztott sávszélességű folyam sávszélességét ne csökkentenénk
            - „Jómódú nem gazdagodhat tovább a szegényebbek rovására, de szegény gazdagodhat jómódú rovására.”



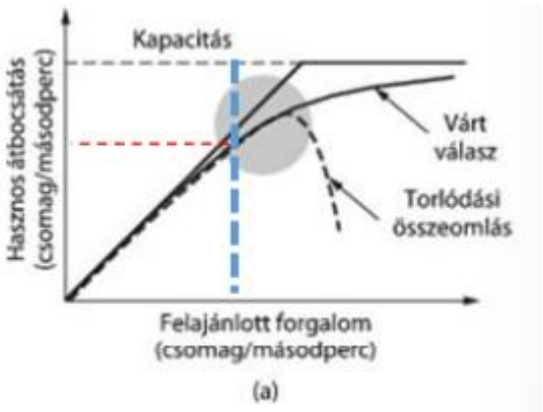
Maximum-minimum sávszélesség-kiosztás négy folyamra
A példában minden adatkapcsolat egységnyi sávszélességű

    c. Konvergencia
        i. Gyorsan konvergáljon egy igazságos és hatékony sávszélesség-kiosztás felé
        ii. Az igények gyakran változnak

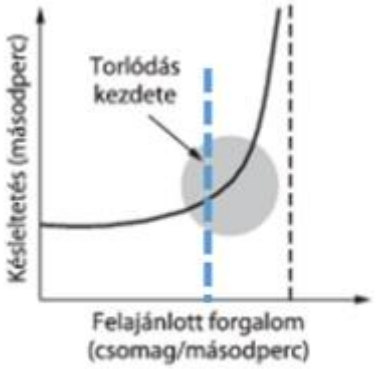


d. Visszacsatolás (érzékelés)
    i. Hogyan észleljük a torlódást? Milyen visszajelzést alkalmazzunk?
    ii. Explicit/implicit
        - Explicit: a torlódásról egyértelmű jelzést adunk (pl. ECN)
        - Implicit: a torlódást egyéb jelekből „sejtjük” (pl. körülfordulási idő növekedése)
    iii. Pontos/pontatlan
        - Pontos: útválasztók a küldési sebességet pontosan meghatározzák a küldő számára (pl. XCP)
        - Pontatlan: a küldési sebességet megpróbáljuk beszabályozni (de nem tudjuk az elvárt értéket)
            a. Ha van torlódás: sebesség csökkentése
            b. Ha nincs torlódás: sebesség növelése

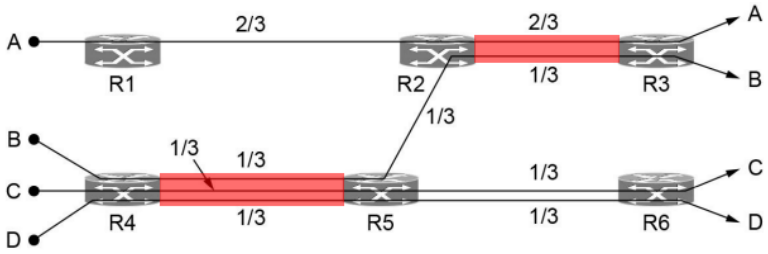
35. Ismertesse a forgalomszabályozás és a torlódáskezelés feladatait. Mi a különbség a két fogalom között?
    a. Fogalmak feljebb
    b. A forgalomszabályozás a hálózati kommunikációban fogadó felet játszó eszközt védi a túlterheléstől, a kommunikáló felek közötti kapcsolatot felügyeli és szabályozza, ezeknél az eszközöknél dolgozik.
    c. A torlódáskezelés a teljes hálózatot és annak infrastruktúráját védi a túlterhelés ellen, routereket, switchek, hálózati végpontoknál dolgozik.
36. Rajzolja fel a hasznos átbocsátást a felajánlott forgalom függvényében. Mutassa meg a torlódás hatását.



37. Rajzolja fel a hálózat késleltetését a felajánlott forgalom függvényében. Mutassa meg a torlódás hatását.



38. A 30. fólián látható hálózatban az A állomásból induló folyam sávszélességét 0.5-re állítjuk. Maximum-minimum igazságosság-e ez a felosztás? Miért?



Maximum-minimum sávszélesség-kiosztás négy folyamra
A példában minden adatkapcsolat egységnyi sávszélességű

a. Mindegyik vonal teljes kapacitása 1 egységet ér (R1 → R2 → R3 link teljes kapacitása 1 egység, R4 → R5 → R6 link teljes kapacitása szintén 1 egység)
b. Ha A 0,5 egységet foglal az R1 → R2 → R3 linken, akkor a maradék kapacitás=1-0.5=0.5.
c. Ezt a maradék 0.5 a B kapja, mert B folyam az R1 → R2 → R3 linken osztozik A-val
d. R4 → R5 → R6 link kapacitása szintén 1. Ezen osztozik C és D. Ez testvérek között is 0.5-0.5 C-nek és D-nek.
e. Így mindegyik folyam 0.5 egységgel rendelkezik, így ez maximum minimum igazságos.

39. Milyen alapvető szabályozási törvényt alkalmazunk a torlódáskezelés során?
    a. Alaptörvény („alkotmány”):
        i. Ha van torlódás: sebesség csökkentése
        ii. Ha nincs torlódás: sebesség növelése
    b. De mi módon csökkentsük/növeljük a sebességet?
        i. Additív: a sebességet egy állandó mennyiséggel változtatjuk (pl. +1)
            - Igazságossági egyenessel párhuzamosan mozog
        ii. Multiplikatív: a sebességet egy állandó mennyiséggel szorozzuk (pl. × 0.5)
            - Origóból induló egyenes mentén mozog
    c. Igazságossági egyenes
        i. Ennek mentén igazságos az elosztás
    d. Hatékonysági egyenes
        i. Ennek mentén hatékony az eloszt
        ii. itt használjuk ki a teljes sávszélességet
40. Ismertesse az AIMD szabályozási törvényt. Illusztrálja ennek működését két állomás esetén.
    a. Additive Increase Multiplicative Decrease
    b. A növelés legyen additív
    c. A csökkentés legyen multiplikatív
    d. Az optimális pontba konvergál
        i. tehát optimális és hatékony is lesz
41. Miért nem okoz gondot a vezeték nélküli hálózatokban gyakori csomagvesztés a torlódás érzékelésében (akkor sem, ha az érzékelés csomagveszteségen alapul)?
    a. A csomagvesztést a szállítási réteg arra használja, hogy a torlódást jelezze
    b. De a vezeték nélküli hálózatokban mindenképpen nagyon sok csomag elveszik
        i. Ez nem a torlódás miatt van
        ii. A vezeték nélküli adatkapcsolati réteg érzékeli a keretvesztést
            - Azonnal javít: újraküld néhány μs-on belül
        iii. Az átviteli hiba gyorsan javításra kerül, a szállítási réteg nem veszi észre
            - A szállítási rétegben az időzítők a ms … s nagyságrendben vannak

1. Ismertesse az IPv4 fejrész egyes mezőinek feladatát (ábra alapján).
    a. Verzió: Az IP protokoll verzióját határozza meg (IPv4 esetén ez 4).
    b. IHL (Header Length): A fejrész hossza 32 bites szavakban.
    c. Differenciált szolgáltatások:
        i. 6 bit: szolgáltatási osztályok (pl. gyorsított, biztosított)
        ii. 2 bit: explicit torlódásértesítés
    d. Teljes hossz: A teljes IP csomag hossza (fejrész + adatrész).
    e. Azonosítás:
        i. A csomag darabolásához szükséges azonosító.
        ii. Azonos datagramhoz tartozó darabok azonosítója ugyanaz.
    f. DF/MF: Darabolás tiltása (DF) vagy a darabok folytatásának jelzése (MF).
    g. Darabeltolás:
        i. A csomagrész helye a datagramban.
    h. Élettartam (TTL):
        i. A csomag maximális ugrásszáma.
        ii. minden ugrásnál értéke csökken eggyel
        iii. Ha nulla, a csomagot el kel dobni
        iv. Megelőzi, hogy hiba esetén a csomagok végtelen ideig kószáljanak
    i. Protokoll: A szállítási protokoll (pl. TCP vagy UDP).
    j. Fejrész ellenőrző összeg:
        i. A fejrész hibáinak ellenőrzése.
        ii. 16 bites ellenőrző összeg
        iii. Minden ugrásnál számítani kell (hiszen az élettartam változik)
    k. Forrás- és célcím: 32 bites IPv4 címek.
    l. Opciók:
        i. Azonosító, hossz, adat
        ii. Ritkán használt funkciók, például útvonalválasztás megadása.
    m. Nem használt bit

2. Ismertesse az IPv4 címek felépítését. Mit jelent a prefix, a hoszt-rész, az alhálózati maszk és a prefix hossz?
    a. Prefix: A hálózati rész azonosítására szolgáló bitsorozat.
    b. Host-rész: Az állomások azonosítására használatos címrészek.
    c. Alhálózati maszk: Az egyesek a hálózati részt, a nullák a hoszt részt jelölik.
    d. Prefix hossz: Az alhálózati maszkban található egyes bitek száma.

3. A 193.44.73.12/26 IP-címhez határozza meg a cím bináris alakját, a bináris és decimális alhálózati maszkot, a hálózat címét, az első és utolsó hoszt címet és a broadcast-címet. Hány állomás lehet ezen a hálózaton?
    a. **Bináris cím**: 11000001.00101100.01001001.00001100.
    b. **Bináris alhálózati maszk**: 11111111.11111111.11111111.11000000.
    c. **Decimális maszk**: 255.255.255.192.
    d. **Hálózat címe**: 193.44.73.0.
    e. **Első hoszt**: 193.44.73.1.
    f. **Utolsó hoszt**: 193.44.73.62.
    g. **Broadcast cím**: 193.44.73.63.
    h. **Hosztok száma**: $2^{(32-26)}-2=62$.

4. Mit jelent az IPv4-en az unicast, broadcast és multicast? Hogyan működnek?
    a. Unicast: Egyetlen címzett.
    b. Broadcast: Az adott hálózat minden eszközének küldött üzenet.
    c. Multicast: Egy adott csoporthoz tartozó hosztok számára küldött üzenet.

5. Ismertesse a privát IP-címek szerepét és használatát az IPv4-ben.
    a. Hálózaton belüli címzésre
    b. Az IPv4 privát címek NAT segítségével érhetnek el külső hálózatokat, így a publikus IP-címek fogyasztását csökkentik.

6. Mit jelent a loopback-cím, mire használjuk?
    a. Öncímző
    b. Címzett: a küldő hoszt
    c. Tesztelésre használjuk

7. Mire szolgálnak az autokonfigurációs címek?
    a. Amikor egy eszköz nem kap IP-címet DHCP-szervertől, automatikusan választ egy címet a megadott tartományból.
    b. Az eszköz ezen a címtartományon belül **ARP-protokollal** ellenőrzi, hogy a választott cím egyedi-e
    c. Csak **helyi hálózaton belüli kommunikációra** használható, mivel ezek a címek nem routolhatók, tehát nem léphetnek ki az adott hálózatról
    d. Pl. Windows DHCP kliens használja, ha nincs elérhető DHCP szerver

8. Ismertesse az alhálózatok létrehozásának módszerét egy közös címtartományon belül.
    a. Egy rendelkezésre álló címblokk további részekre (alhálózatokra) bontható
    b. A bitkölcsönzés módszerét használjuk az alhálózati címek kialakítására. Ehhez a hoszt rész biteiből kölcsönzünk biteket, hogy az alhálózati címeket azonosítsuk.
    c. Az alhálózatok száma: 2n2^n2n, ahol nnn a kölcsönzött bitek száma.
    d. Egy alhálózatra kioszthatő hosztok száma: $2^{32\text{-alhlázótai prefix}}$–2 (a hálózati és broadcast cím miatt)

9. Egy vállalatnak a 128.129.130.0/24 IPv4 címtartomány áll rendelkezésére. Szeretnénk két egyforma méretű alhálózatot létrehozni (Fejlesztési osztály [FO] és Adminisztráció [AD]). Mi lesz a FO alhálózati címe és alhálózati maszkja? Mi lesz az AD alhálózati címe és alhálózati maszkja? Hány állomás lehet az egyes alhálózatokon?
    a. /24-es tartomány két /25-ösre bontható, mert 2 alhálózat az 21: /24+1=/25
    b. FO:
        i. alhálózati címe: 128.129.130.0/25
        ii. alhálózati maszkja: 255.255.255.128
        iii. állomások száma: 32-25=7 => 27=128 => 128-2=126
        iv. a .0 lesz a hálózati cím, a .127 pedig a broadcast cím, ezeket nem lehet kiosztani ezért 2-t le kell vonni, hogy megkapjuk a ténylegesen kioszthatő címek számát
    c. AD:
        i. alhálózati címe: 128.129.130.128/25
        ii. alhálózati maszkja: 255.255.255.128
        iii. állomások száma: 32-25=7 => 27=128 => 128-2=126
        iv. a .128 lesz a hálózati cím, a .255 pedig a broadcast cím, ezeket nem lehet kiosztani ezért 2-t le kell vonni, hogy megkapjuk a ténylegesen kioszthatő címek számát

10. Egy vállalatnak a 128.129.131.0/24 IPv4 címtartomány áll rendelkezésére. Szeretnénk a címtartomány felét a Fejlesztési osztálynak [FO], negyedét az Adminisztrációnak [AD]), negyedét pedig a Kereskedelmi Osztálynak [KO] létrehozandó alhálózatokhoz rendelni. Mi lesz a FO alhálózati címe és alhálózati maszkja? Mi lesz az AD alhálózati címe és alhálózati maszkja? Mi lesz a KO alhálózati címe és alhálózati maszkja? Hány állomás lehet az egyes alhálózatokon?
    a. **128.129.131.0/24** (256 cím, 2^8)
    b. FO (Fejlesztési osztály)
        i. A tartomány fele (128 cím), azaz /25
        ii. Hálózati cím: **128.129.131.0/25**
        iii. Maszk: **255.255.255.128**
        iv. Broadcast cím: **128.129.131.127**
        v. Hosztok száma: **126**
    c. AD (Adminisztráció)
        i. A tartomány negyede (64 cím), azaz /26
        ii. Hálózati cím: **128.129.131.128/26**
        iii. Maszk: **255.255.255.192**
        iv. Broadcast cím: **128.129.131.191**
        v. Hosztok száma: **62**
    d. KO (Kereskedelmi osztály)
        i. A tartomány negyede (64 cím), azaz /26
        ii. Hálózati cím: **128.129.131.192/26**
        iii. Maszk: **255.255.255.192**
        iv. Broadcast cím: **128.129.131.255**
        v. Hosztok száma: **62**

11. A fenti vállalathoz érkezik a 128.129.131.173 címre egy csomag. Hogyan kell eldönteni, hogy melyik alhálózatra továbbítsuk?
    a. A csomag cél IP-címét az alhálózati maszk segítségével "maszkoljuk" és ellenőrizzük, hogy melyik alhálózat hálózati címe egyezik az eredménnyel
    b. Ez az **útválasztás (routing)** alapja
    c. **Példa maszkolásra**:
        i. Célcím binárisan: 10000000.10000001.10000011.10101101 (128.129.131.173).
        ii. **AD alhálózat maszkja**: 11111111.11111111.11111111.11000000 (255.255.255.192).
        iii. **Maszkolás eredménye**: 10000000.10000001.10000011.10000000 (128.129.131.128).

iv. Az eredmény egyezik az AD alhálózat hálózati címével, ezért ide tartozik

12. Mit jelent az előtagok csoportosítása? Milyen előnnyel jár?
    a. olyan technika, amelyet az IP-címek kezelésére használnak
    b. több IP-címet, amelyek logikailag közel állnak egymáshoz (azaz ugyanabban a hálózati tartományban vagy alhálózatban találhatók), egyetlen nagyobb IP-címblokká egyesítenek
    c. célja a hálózati címek hatékonyabb és egyszerűbb kezelése, valamint az internetes routing táblák optimalizálása
    d. CIDR (Classless Inter-Domain Routing) eljárás segítségével végzik
    e. Előnyei:
        i. Kevesebb routing bejegyzés
        ii. Hatékonyabb hálózati forgalom
        iii. Jobb IP-címgazdálkodás
        iv. Jobb hálózati skálázhatóság

13. Hogyan működik a Leghosszabb egyező előtag útválasztás?
    a. Csomag beérkezik, a címe A
    b. Útválasztó megnézi, hogy az A cím melyik bejegyzésére illeszkedik
    c. Ha több ilyen bejegyzés is van, akkor kiválasztja azt, amelyik a leghosszabb előtaggal bír
        i. Pl. ha egy /22 és egy /19 bejegyzésre is illeszkedik, akkor a /22 bejegyzést használja
    d. A bejegyzésnek megfelelő irányba küldi tovább a csomagot

14. Egy útválasztóban az alábbi bejegyzések találhatók: 194.24.0.0/19 à London, 194.24.0.0/25 à Gézaháza-alsó. Hová kell továbbítani a 194.24.0.78 címre érkező üzenetet? És a 194.24.0.176 címre érkező üzenetet?
    a. leghosszabb előtag egyezés: a router mindig azt a bejegyzést választja, amelyiknek a leghosszabb előtagja (prefixe)
    b. 194.24.0.0/19:
        i. Hálózati cím: 194.24.0.0
        ii. Első cím: 194.24.0.1
        iii. Utolsó cím: 194.24.31.254
        iv. Szórási (broadcast) cím: 194.24.31.255 => mert 2^13=8192/256=32 (.0.0-.31.255)
        v. Az **194.24.0.78** benne van ebben a tartományban.
    c. 194.24.0.0/25:
        i. Hálózati cím: 194.24.0.0

ii. Első cím: 194.24.0.1
iii. Utolsó cím: 194.24.0.126
iv. Szórási (broadcast) cím: 194.24.0.127
v. Az **194.24.0.78** benne van ebben a tartományban.7

d. **Leghosszabb előtag egyezés**:
i. Mindkét tartomány lefedi az IP-címet, de a **/25-ös előtag** hosszabb, mint a /19-es, ezért az **194.24.0.78** címre érkező csomagot **Gézaháza-alsóra** kell továbbítani.
ii. Az **194.24.0.176** cím csak a /19-es előtaggal egyezik, ezért ezt a csomagot **London** felé kell továbbítani

15. Mit jelent a CIDR?
a. **Classless Inter-Domain Routing** (**Osztály nélküli körzetek közötti útválasztás)**
b. Nincsenek előre meghatározott előtag-méretek
c. hatékony címkiosztás

16. Miért van szükség NAT-ra? Ismertesse a NAT működését.
a. **NAT** (Network Address Translation) alapvetően a hálózatok és az internet címzési problémáinak kezelésére szolgál
b. Az IP-címek korlátozott száma miatt vált szükségessé
c. Biztonság:
i. elrejti a belső hálózat eszközeit a külső hálózat (internet) felől
d. Hálózatok összekapcsolása:
i. Lehetővé teszi különböző belső hálózatok számára, hogy ugyanazokat a privát IP-tartományokat használják, miközben egy közös nyilvános IP-címen keresztül kommunikálnak
e. Működése:
i. A belső hálózati eszközök (például számítógépek, telefonok) privát IP-címekkel rendelkeznek, amelyek nem elérhetők az interneten
ii. Amikor egy belső eszköz kérést küld az internetre (pl. egy weboldal megnyitásakor), az útválasztó:
- Lecseréli a csomag forráscímét a nyilvános IP-címére.
- Megjegyzi az eredeti forráscímet és a hozzá tartozó portot egy fordítási táblában.

iii. A válaszcsomag megérkezésekor az útválasztó:
- Megnézi a fordítási táblát.
- Visszacseréli a célcímet az eredeti privát IP-címre, majd továbbítja a belső hálózatra.

17. Egy NAT-táblában a következő bejegyzések találhatók (fiktív port, IP-cím, port):

    6783 192.168.54.12 8080
    9845 192.168.54.12 1980
    1231 192.168.54.54 8080

    Egy bejövő üzenet címe a következő: 173.67.86.24:1231. Hová kell továbbítani az üzenetet? Mi a 173.67.86.24? Mi a 1231?

    a. **173.67.86.24**:
    Ez a külső (nyilvános) IP-cím, amelyet a NAT-tábla kezel. Ez általában az útválasztó nyilvános IP-címe, amelyen keresztül az internethez csatlakozik.
    b. **1231**:
    Ez a célport, amely alapján a NAT-tábla meg tudja határozni, hogy az üzenetet melyik belső eszköznek (IP-cím és port) kell továbbítania. A portok a kommunikáció azonosítására szolgálnak a fordítási táblában.
    c. A bejövő üzenet címe: **173.67.86.24:1231**.
        i. A NAT-tábla alapján a **1231** külső port hozzárendelve van a következő belső címhez:
            - **Belső IP-cím: 192.168.54.54**
            - **Belső port: 8080**
            - Ez azt jelenti, hogy az üzenetet továbbítani kell a **192.168.54.54** belső IP-címre, az **8080-as porton**.

18. Miért nem univerzális megoldás a NAT?
    a. IP címek nem globálisan egyediek
        i. pl. sok 10.0.0.1 cím a hálózaton
    b. Nem tud bárki bárkinek küldeni
        i. NAT mögül csak kezdeményezni lehet
        ii. További trükkök kellenek ennek áthidalására
    c. Az összeköttetés nélküli internetbe összeköttetés-alapú kapcsolatot kever
        i. NAT tábla sérülése: összeomlás
    d. Összemossa a 3. és 4. réteget
        i. portszámok: 4. réteg
    e. TCP-n és UDP-n kívül más szállítási protokollok is vannak

19. Mik az IPv4 és IPv6 közötti legfőbb különbségek?
    a. Sokkal több IP cím van (32 bites helyett 128 bites cím)
    b. Fejrész egyszerűbb lett (13 mezőből 7 lett)
    c. Opciók jobb támogatása

i. Szükségszerű a rövidebb fejrész miatt
ii. Gyorsabb csomagfeldolgozást tesz lehetővé

d. Biztonság javítása
e. Szolgáltatásminőség nagyobb hangsúlyt kapott
i. Multimédia

20. Ismertesse az IPv6 fejrész egyes mezőinek feladatát (ábra alapján).
a. Verzió:
i. 6
ii. Ez a mező az IPv6 verzióját jelzi
b. Differenciált szolgáltatások:
i. Torlódásjelzés (ECN – Explicit Congestion Notification):
- Lehetővé teszi a hálózati eszközök számára, hogy jelezzenek a hálózat terheltségéről, így a forgalom irányításában figyelembe vehetjük a hálózati állapotot.

ii. A hálózati csomagok különböző szintű prioritással való kezelése, hogy a késleltetés és a sávszélesség alakulása alapján fontosabb forgalmakat előnyben részesíthessünk
c. Folyamcímke:
i. Virtuális-áramkör alapú megközelítést tesz lehetővé
ii. Folyamot előre fel lehet állítani, ez az azonosítója
iii. Minden útválasztó kikeresi táblázatából, hogy a címke milyen különleges elbánást igényel
d. Adatmező hossza:
i. Ez a mező az adatmező hosszát tartalmazza (az IPv6 fejléc után), és meghatározza, hogy hány bájt adat található a csomagban. Az adatmező (payload) minden olyan információt tartalmaz, amely nem része a fejlécnek, például a szállítási protokollok adatcsomagjai (TCP, UDP, ICMPv6 stb.).
e. Következő fejrész:
i. Jelzi, hogy van-e következő opcionális fejrész, és milyen típusú
ii. Ha ez az utolsó IP fejrész, akkor itt jelzi, hogy melyik szállítási protokollnak kell a csomagot adni (TCP, UDP)
f. Ugráskorlát:
i. Mint az IPv4 élettartam
g. Forráscím, célcím
i. 128 bit (16 bájt)
ii. Formátum:
- 8 csoport

- Kettősponttal elválasztva
- Mindegyik csoportban 4-4- hexadecimális számjegy (hextet)
- Pl.: 8000:0000:0000:0000:0123:4567:89AB:CDEF

iii. Egyszerűsítések a jelölésben:
- Csoporton belül a bevezető 0-k elhagyhatók
  a. Pl.: 0123 → 123
- Csupa nulla csoportok (egy vagy több) két kettősponttal helyettesíthető
  a. csak egyszer egy címben
  b. Pl.: 8000::123:4567:89AB:CDEF
- IPv4 címek írásmódja:
  a. ::192.31.20.46

21. Mit jelent az IPv6-ban az unicast, multicast és anycast? Hogyan működnek? Mit jelent a GUA és LLA?
    a. Unicast
        i. Egyetlen címzett
        ii. Globális (Global Unicast Address - GUA):
            - mint IPv4, globálisan egyedi. (Opcionális)
            - 2000::/3 (első 3 bit: 001)
        iii. Lokális (Link-Local Address - LLA):
            - csak helyi hálózatra alkalmazzuk. (Kötelező)
            - fe80::/10
        iv. Localhost:
            - ::1/128
    b. Multicast
        i. Több címzett, prefix: ff00::/8
        ii. Pl.: ff02::1 – minden hoszt, ff02::2 – minden router (azonos adatkapcsolaton)
    c. Anycast
        i. Unicast üzenet (egy cím), de a címzett bármely lehet a lehetséges címzettek közül (általában a legközelebbi)
22. Hogyan lehet IPv6 alatt alhálózatokat létrehozni? Milyen részekre tagozódik az IPv6 cím?
    a. Van elég bit (128)
        i. Hoszt címbitek: 64 bit
        ii. Prefix: összesen 64 bit – alhálózatok meghatározásához kell
            - Ebből a 16 alsó bit az alhálózatokat különbözteti meg

- A 48 felső bit a globális útvonalválasztási előtag

23. Ismertesse az IPv6 címek felépítését, jelölését.
    a. 128 bit: Az IPv6 címek teljes hossza 128 bit.
    b. A cím nyolc darab 16 bites hextet-re van bontva, mindegyik 4 hexadecimális számjegyből áll.
    c. pl.: 2001:0db8:0000:0042:0000:8a2e:0370:7334
    d. **IPv6 prefix** tartalmazhat globális (pl. 2000::/3), link helyi (pl. fe80::/10) vagy privát (pl. fc00::/7) címeket
24. Írja fel teljes alakban (rövidítés nélkül) a következő IP-címeket. Azonosítsa a címekben az útválasztási előtagot, az alhálózat azonosítóját és a hoszt interfész azonosítóját.
    a. Teljes alakra hozás:
        i. rövidített alaknál a felesleges 0-ák el vannak hagyva
        ii. több 0 csoportot ::-al kiváltunk (csak egyszer)
        iii. Így vissza lehet fejteni a címet azzal hogy ahol nem 4 elem van a hextetben oda beírjuk a 0-kat.
    b. Útválasztási előtag (prefix) meghatározása:
        i. Az IPv6 címben az útválasztási előtagot (prefixet) a cím elején lévő hálózati rész határozza meg, amelyet általában egy perjellel (/) jelzett számmal adnak meg.
            - Példa: 2001:db8::/32. Ez azt jelenti, hogy az első 32 bit a hálózati címhez tartozik.
        ii. A prefixet hexadecimális blokkonként kell értelmezni:
            - Minden hexadecimális számjegy 4 bitet jelent.
            - Példa: /32 → az első 8 hexadecimális számjegy (2001:0db8) a prefix.
        iii. Alhálózat azonosítójának meghatározása:
            - Az IPv6 alhálózati azonosítóját a prefix hosszán túli bitmezők tartalmazzák, egészen az első 64 bitig.
                a. Példa: ha a prefix /48, akkor az alhálózat azonosítója a következő 16 bit (a 49–64. bit).
                b. Az alhálózat azonosítóját a cím teljes alakjában azonosítjuk
                c. Ha nincs /szám akkor /64-et feltételezünk
        iv. Hoszt interfész azonosítójának meghatározása
            - Az hoszt interfész azonosítója a cím 64 biten túli része, vagyis az utolsó 64 bit.
                a. Ez az azonosító a hosztot egyértelműen megkülönbözteti az alhálózaton belül.
                b. Példa: 0001:0000:0000:0012 az utolsó 64 bitben

c. 8000:1::123:67:89AB:CDE
    i. Teljes alak: 8000:0001:0000:0000:0123:0067:89AB:0CDE
    ii. Útválasztási előtag (Prefix): 8000:0001:0000:0000
    iii. Alhálózat azonosítója: 8000:0001:0000:0000
    iv. Hoszt interfész azonosítója: 0123:0067:89AB:0CDE
d. 2001::1
    i. Teljes alak: 2001:0000:0000:0000:0000:0000:0000:0001
    ii. Útválasztási előtag (Prefix): 2001:0000:0000:0000
    iii. Alhálózat azonosítója: 2001:0000:0000:0000
    iv. Hoszt interfész azonosítója: 0000:0000:0000:0001
e. 2001:db8:12:567:1::12
    i. Teljes alak: 2001:0db8:0012:0567:0001:0000:0000:0012
    ii. Útválasztási előtag (Prefix): 2001:0db8:0012:0567
    iii. Alhálózat azonosítója: 2001:0db8:0012:0567
    iv. Hoszt interfész azonosítója: 0001:0000:0000:0012
f. fe80::
    i. Teljes alak: fe80:0000:0000:0000:0000:0000:0000:0000
    ii. Útválasztási előtag (Prefix): fe80:0000:0000:0000
    iii. Alhálózat azonosítója: fe80:0000:0000:0000
    iv. Hoszt interfész azonosítója: 0000:0000:0000:0000
g. ::1
    i. Teljes alak: 0000:0000:0000:0000:0000:0000:0000:0001
    ii. Útválasztási előtag (Prefix): 0000:0000:0000:0000
    iii. Alhálózat azonosítója: 0000:0000:0000:0000
    iv. Hoszt interfész azonosítója: 0000:0000:0000:0001
h. 2001:2:3:4:5:6:7:8
    i. Teljes alak: 2001:0002:0003:0004:0005:0006:0007:0008
    ii. Útválasztási előtag (Prefix): 2001:0002:0003:0004
    iii. Alhálózat azonosítója: 2001:0002:0003:0004
    iv. Hoszt interfész azonosítója: 0005:0006:0007:0008

25. Sorolja fel az IPv4 és IPv6 vezérlő protokolljait. Ismertesse a szerepüket.
    a. IPv4
        i. ICMP - Internet Control Message Protocol
            - Internetes vezérlőüzenet protokoll
            - Váratlan események jelzésére, tesztelésre
            - Figyelmeztet a hibás útvonalválasztásra vagy elérhetetlen célállomásra
        ii. ARP - Address Resolution Protocol
            - Címfeloldási protokoll
            - Az IPv4 címekhez tartozó MAC-címek (fizikai címek) lekérdezésére szolgál a helyi hálózatban
        iii. DHCP - Dynamic Host Configuration Protocol
            - Dinamikus hosztkonfigurációs protokoll (IP-cím, Subnet-mask, DNS szerver, Alapértelmezett átjáró)
    b. IPv6
        i. ICMPv6
            - ICMP-hez hasonló funkciók
                a. Hiba, visszhang, időtúllépés stb jelzése
26. Ismertesse az ICMP(v4) legfontosabb üzenettípusait. Hogyan működik és milyen üzeneteket használ a ping és tracert?
    a. Üzenetek:
        i. Cél elérhetetlen: Csomagot nem lehetett kézbesíteni
        ii. Időtúllépés: Az Élettartam mező elérte a 0-t
        iii. Paraméter probléma: Érvénytelen fejrész mező
        iv. Forráslefojtás: Lefojtócsomag
        v. Átirányítás: Egy útválasztót tanít meg a földrajzra
        vi. Visszhang kérés/válasz: Ellenőrzi, hogy életben van-e a gép
        vii. Időbélyeg kérés/válasz: Ugyanaz, mint a visszhang csak időbélyeggel
        viii. Útválasztó kérelmezés/hirdetés: Egy közeli útválasztó megtalálása
    b. ping: az elérhetőség ellenőrzése, Visszhang kérés/választ használ
    c. tracert: az útvonal elemzése a csomag célba juttatása során, Időtullépést használja

27. Ismertesse az ARP üzeneteit és a protokoll működését.
    a. Probléma:
        i. Hálózati réteg logikai (IP) címeket használ
        ii. Hálózati réteg az adatkapcsolati réteg szolgáltatásait használja
        iii. De az adatkapcsolati réteg fizikai (MAC) címeket használ
        iv. Honnan tudjuk, hogy melyik fizikai címhez melyik logikai cím tartozik?
    b. Az ARP ezeket kezeli
        i. ARP REQUEST:
            - Broadcast keret küldése az adatkapcsolati rétegben:
        ii. ARP REPLY:
            - Válasz a feladónak (unicast)
    c. Üzenet küldése:
        i. Ethernet keret felépítés (forrás: AAA, cél: BBB)
        ii. Üzenet elküldése
    d. Üzenet vétele:
        i. Ethernet keretet veszi a BBB című állomás
        ii. Keretet leveszi, a csomagot átadja az hálózati rétegnek.
    e. ARP használata:
        i. Hálózaton belüli cím: másik hoszt fizikai címe
        ii. Hálózaton kívüli cím: átjáró fizikai címe
28. Ismertesse az ARP táblázat szerepét és használatát.
    a. Optimalizálás:
        i. ARP táblázat
        ii. Ismert címeket tartalmazza:
            - IP - MAC
        iii. Idővel „lejár”
        iv. Működés:
            - Küldés előtt ellenőrzi a táblát
                a. Ha van bejegyzés, használja
                b. Ha nincs, ARP REQUEST
            - ARP REPLY eredményét beírja

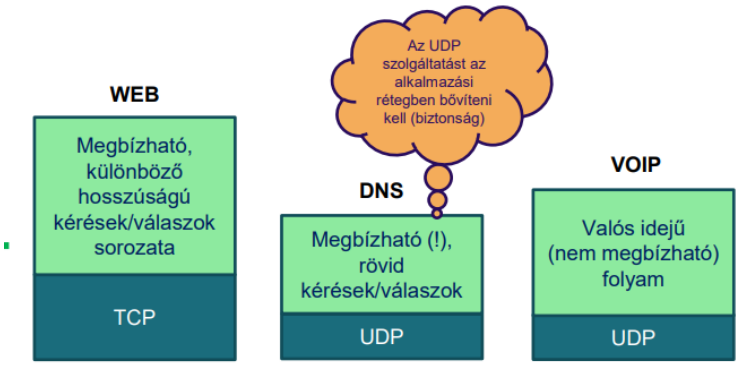
29. Mi célt szolgál a DHCP? Ismertesse a DHCP üzeneteit és a protokoll működését. Lehet-e DHCP nélkül működtetni egy hálózatot. Ha nem, akkor miért, ha igen, akkor hogyan?
    a. IP címeket „lízingel”: meghatározott időre
    b. Lehetővé teszi, hogy az eszközök manuális konfiguráció nélkül csatlakozzanak a hálózathoz
    c. IP cím kérés:
        i. DHCP DISCOVER: Broadcast kérés
            - Felméri van-e DHCP szerver és IP címet kér tőle
        ii. DHCP OFFER: Unicast válasz
            - Felajánl egy IP címet
        iii. DHCP REQUEST: Broadcast kérés
            - Elfogadja az IP címet
        iv. DHCP ACK: Unicast válasz
            - Meghatározza mennyi ideig használhatja a kapott IP címet (lízing)
    d. IP cím frissítése (lízing lejárta előtt):
        i. DHCP REQUEST: Broadcast kérés
            - Ismét kérné a korábban kért címet
        ii. DHCP ACK: Unicast válasz
            - Ismét megkapja meghatározott ideig
    e. Lehet DHCP nélkül is üzemeltetni hálózatot, viszont akkor manuálisan kell kiosztani a címeket, subnet maszkot, alapértelmezett átjárót, DNS kiszolgálót az eszközöknek (Statikus IP)
30. Mire szolgál az ICMPv6? Mire szolgál a Neighbor Discovery Protocol (ND)?
    a. Hiba, visszhang, időtúllépés stb. jelzésére szolgál
    b. Az ND az IPv6 hálózatokban használt alapvető protokoll, amely a helyi hálózaton található szomszédos eszközök felfedezésére és kommunikációjának kezelésére szolgál

31. Ismertesse a NS és NA üzenetek szerepét, a szomszédságfelderítés menetét.
    a. Neighbor Discovery protocol (ND)
        i. Neighbor Solicitation (NS, szomszéd megszólítása)
        ii. Neighbor Advertisement (NA, szomszéd hirdetés)
        iii. Router Solicitation (útválasztó megszólítása)
        iv. Router Advertisement (útválasztó hirdetés)
    b. Szomszédság felderítés (ND=NS+NA)
        i. Szeretnék egy IPv6 címre üzenetet küldeni.
        ii. Tudom-e a MAC címét?
            - Neighbor Cache: mint ARP tábla
            - Ha van benne info, akkor használjuk (GOTO 5), ha nincs, akkor GOTO 3
        iii. NEIGHBOR SOLICITATION (NS): Multicast keret az adatkapcsolati rétegben:
            - Körbe küld egy IP címet, és megkérdezi, hogy kié
        iv. 4. NEIGHBOR ADVERTISEMENT (NA):
            - Jelzi, hogy övé az IP cím
        v. Üzenet küldése
        vi. Üzenet vétele
            - Neighbor cache frissítése
32. Ismertesse az RS és RA üzenetek használatát. Milyen információt hordoz az RA üzenet? Milyen módokon juthat egy hoszt IPv6 címhez?
    a. Neighbor Discovery protocol (ND)
        i. Neighbor Solicitation (NS, szomszéd megszólítása)
        ii. Neighbor Advertisement (NA, szomszéd hirdetés)
        iii. Router Solicitation (útválasztó megszólítása)
        iv. Router Advertisement (útválasztó hirdetés)
    b. IPv6 DHCP-szerű funkciója a globális IP-cím (GUA) előállítására
    c. ROUTER SOLOCITATION (RS): Multicast üzenet minden útválasztónak
        i. Az üzenetben egy bizonyos címről kér le információkat
    d. ROUTER ADVERTISEMENT (RA): Multicast üzenet minden állomásnak
        i. Válaszban küldi a címről a tudnivalókat
            - Hálózati prefix és a prefix hossza
            - Alapértelmezett átjáró címe
            - DNS címe
            - [DHCP szerver címe]
        ii. De mi legyen a hoszt cím (a prefix már megvan)?

- Csinálj magadnak egyet (SLAAC)
- Küldök egy DHCP szerver címet, konzultálj vele

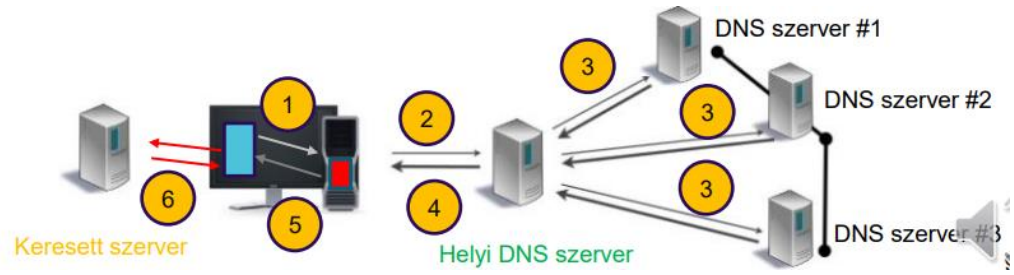
33. Mit jelent a SLAAC? Ismertesse a működését.
    a. SLAAC (Stateless Address Autoconfiguration)
    b. IPv6 hálózati címek automatikus konfigurációs módszere
    c. lehetővé teszi az eszközök számára, hogy önállóan konfigurálják az IPv6-címüket
    d. a hálózaton belüli routerek segítségével történik, és a címeket dinamikusan generálja, a hálózati prefixekből és az eszköz MAC-címéből
    e. Működése:
        i. Router Discovery
            - Csatlakozik egy IPv6 hálózathoz az eszköz
            - Küld egy RS (Router Solicitation) jelet, hogy felfedezze a routereket
            - A routerek RA (Router Advertisement) választ küldenek, ami tartalmazza a hálózati prefixet és a konfigurációs információkat.
        ii. RA megküldte a prefixet (64 bit)
            - Hiányzik még a hoszt cím (64 bit)
                a. Véletlenszám (Windows 10)
                b. EUI-64 (Cisco routerek)
                c. Az eszközök ezt a prefixet és a saját **64 bites** hosszú **interfész azonosítót** kombinálják az IPv6-cím létrehozásához
                d. Az interfész azonosítót gyakran az eszköz **MAC-címéből** (a hálózati kártya fizikai címéből) generálják
                e. Ezután ellenőrzik, hogy nem duplikált-e a cím
                    i. Ehhez NS üzenetet küld a többi eszköznek, amelyben a cím felől kérdez
                        1. ha nem duplikált akkor használja
                        2. ha duplikált akkor újat generál

1. Milyen szolgáltatásokat használnak az alkalmazási réteg protokolljai?
    a. A szállítási réteg szolgáltatásait használja
        i. Különféle igények, különféle szolgáltatások



2. Soroljon fel alkalmazási rétegben megvalósított protokollokat. Ismertesse ezek feladatát is.
    a. DNS – Domain Name System
        i. körzetnevek IP-címekké alakítása
    b. E-levelezés
        i. SMTP – Simple Mail Transfer Protocol
            - e-mail küldés
        ii. POP3 – Post Office Protocol
            - e-mail letöltése a szerverről
        iii. IMAP – Internet Message Access Protocol
            - e-mail elérése a szerveren (a szerveren marad)
    c. FTP – File Transfer Protocol
        i. Megbízható fájlátvitel két gép között
    d. HTTP, HTTPS
        i. World Wide Web használata
    e. BitTorrent
        i. P2P fájlmegosztás
3. Mi a DNS, mi a célja?
    a. Az IP hálózati címeket használ (172.217.16.110)
    b. Az emberek a magas szintű neveket kedvelik (google.com)
    c. Névfeloldás:
        i. Magas szintű nevek átalakítása hálózati címekké (google.com → 172.217.16.110)
    d. Általánosabban
        i. Erőforrás-nyilvántartás
            - Magas szintű nevekhez erőforrás-bejegyzések rendelése
            - Pl.: név-szerver, levelező szerver, IPv4 cím, IPv6 cím

4. Mi volt a hosts.txt fájl feladata a korai hálózatokban? Mi a hosts fájl feladata egy mai rendszerben?
    a. ARPANET
        i. A hosts.txt tartalmazta az összes számítógép nevét és IP-címét
        ii. Minden gép éjszakánként frissítette a hosts.txt fájlt
        iii. Egy ideig egész jól működött
    b. Az internet növekedett…
        i. A hosts.txt fájl egyre nagyobb lett
        ii. Központi menedzsment kellett a névütközések elkerülésére
    c. Domain Name System (DNS): 1983
        i. Hierarchikus névkiosztás
        ii. Elosztott adatbázisrendszer
        iii. Összekapcsolja a host-neveket és IP-címeket (és egyéb adatokat): névfeloldás
    d. a hosts fájl továbbra is létezik, szerepe jelentősen csökkent a **DNS** rendszerek bevezetése óta
        i. Helyi névfeloldás biztosítása, amikor egy számítógép gazdanevét kell egy IP-címhez rendelni
5. Ismertesse a DNS működését. Mi a resolver csonk feladata? Mi a helyei DNS szerver feladata?
    a. Címfeloldó eljárás (resolver csonk) meghívása a felhasználói programból (pl. gethostbyname, GetHostEntry, InetAddress.getByName)
    b. A resolver csonk kérést küld a helyi DNS szervernek
    c. A helyi DNS szerver iteratív keresést hajt végre más DNS szerverek hívásával
    d. Helyi DNS szerver visszaküldi az IP címet a resolver csonknak
    e. A resolver csonk visszaadja az eredményt (IP címet) a felhasználói program hívó függvényének
    f. Az IP-cím segítségével a felhasználói program kommunikálhat a keresett szerverrel



6. Mi az ICANN, mi a feladata?
    a. Internet Corporation for Assigned Names and Numbers
    b. Feladata, hogy **koordinálja az internet cím- és névterének működését**, ezzel biztosítva az internet globális és zökkenőmentes működését.
        i. Domain Name System (DNS) kezelése
        ii. IP-címek kezelése
        iii. Protokoll-azonosítók kezelése
        iv. DNS gyökérzóna kezelése
    c. kulcsszerepe, hogy biztosítsa az internet globális elérhetőségét és egységes működését

7. Ismertesse a DNS névtér hierarchikus felépítését.
    a. Elsődleges körzetek
        i. ICANN kezeli
        ii. Eredetileg 6 volt
        iii. Jelenleg több, mint 1000 van
            - Általános: pl.: .com, .edu, .net, .org, …
            - Országok: pl.: .hu, .de, .tv, stb
    b. Másodlagos körzetek
        i. Adminisztrátorok (registrar) kezelik
        ii. Pl.: cisco.com, uni-obuda.hu
    c. Minden körzet maga ellenőrzi az alatta lévő körzetek kiosztását
        i. Pl.: magyarorszag.hu, magyarorszag.org, amk.uni-obuda.hu
    d. Körzeteket pontokkal választjuk el
        i. Balról jobbra a gyökér felé haladunk (a gyökeret nem jelöljük)
        ii. Pl.: fluit.cs.vu.nl, amk.uni-obuda.hu
8. Mit tartalmaz egy DNS bejegyzés (5)?
    a. Körzet-név
    b. Élettartam
        i. érvényességi idő másodpercben
    c. Osztály
        i. általában IN (internet)
    d. Típus
        i. az adat típusa
    e. Érték
        i. a típusnak megfelelő érték
9. Magyarázza el, mit jelent az alábbi DNS bejegyzések:

uni-obuda.hu.	86400	IN	NS	ns1.uni-obuda.hu.
uni-obuda.hu.	86400	IN	A	193.224.41.159
index.hu.	60	IN	AAAA	2a02:730::1860
uni-obuda.hu.	86400	IN	MX	10 sendmail.uni-obuda.hu.
index.hu.	2	IN	SOA	ns.index.hu. support.mail.index.hu. 2018100045 600 300 604800 300

    a. Első sor
        i. Ez egy **NS (Name Server)** rekord, amely meghatározza, hogy az adott domainnév (**uni-obuda.hu**) névszervere a **ns1.uni-obuda.hu**.
        ii. 86400: Az élettartam (Time-to-Live, TTL) másodpercekben, ami azt jelenti, hogy ezt az információt 1 napig (86400 másodpercig) cache-elhetik más DNS-szerverek.
    b. Második sor
        i. Ez egy A rekord, amely azt mondja meg, hogy az uni-obuda.hu domainhez tartozó IPv4 cím a 193.224.41.159.
    c. Harmadik sor
        i. Ez egy AAAA rekord, amely az index.hu domain IPv6 címét mutatja: 2a02:730::1860.

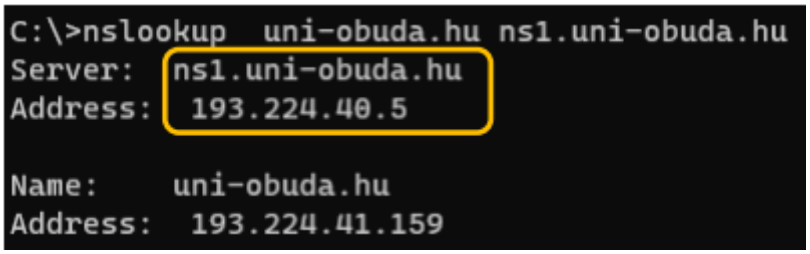
ii. 60: A TTL nagyon alacsony érték, mindössze 60 másodperc, vagyis ezt az adatot gyakran frissítik.

d. Negyedik sor
   i. Ez egy MX (Mail Exchange) rekord, amely megmutatja, hogy az uni-obuda.hu domainhez tartozó e-mail kiszolgáló a sendmail.uni-obuda.hu.
   ii. Az érték 10 a prioritást jelöli, kisebb szám magasabb prioritást jelent.

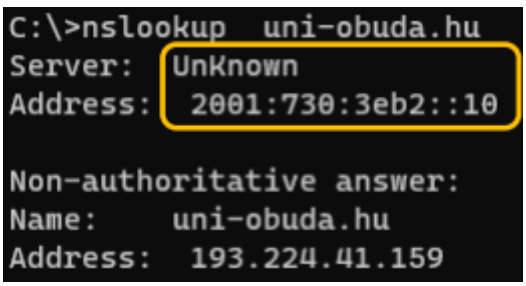
e. Ötödik sor
   i. Ez egy SOA (Start of Authority) rekord, amely egy DNS-zóna adminisztratív adatait tartalmazza.
   ii. ns.index.hu.: A zóna elsődleges névszervere.
   iii. support.mail.index.hu.: Az adminisztratív kapcsolattartó e-mail címe (pont helyett @-ot használunk: support@mail.index.hu).
   iv. 2018100045: A zóna sorszáma (Serial Number), amely a változások követésére szolgál.
   v. 600: Frissítési időköz (Refresh), vagyis az alárendelt névszerverek 600 másodpercenként (10 perc) ellenőrzik, hogy történt-e változás.
   vi. 300: Újrapróbálkozási idő (Retry), ami azt jelenti, hogy ha az első frissítési próbálkozás sikertelen, 300 másodperc után újrapróbálkozik.
   vii. 604800: Lejárati idő (Expire), vagyis 604800 másodperc (7 nap) után az alárendelt szerverek már nem bíznak a zóna adataiban, ha nem tudják frissíteni.
   viii. 300: Minimum TTL, az alapértelmezett élettartam a zóna rekordjaira

10. Mit jelent a hiteles és tárban lévő bejegyzés?

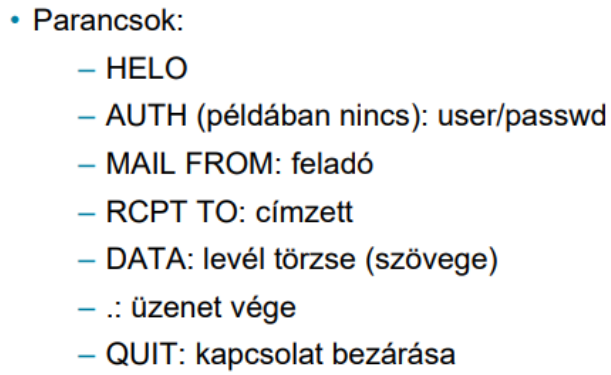
a. Hiteles bejegyzés (authorative record):
   i. Az információ a bejegyzést kezelő szervtől származik (biztosan helyes)



b. Tárban lévő bejegyzés (cached record):
   i. A bejegyzés a gyorstárból való (esetleg idejétmúlt lehet)



11. Automatikus DNS cím beállítása esetén hogyan kapja meg a számítógép a DNS szerver címét?
    a. **DHCP (Dynamic Host Configuration Protocol)** segítségével kapja meg a DNS-szerver címét
    b. A számítógép **DHCP Discover** üzenetet küld a hálózatnak, hogy IP-címet és egyéb konfigurációs adatokat kérjen
    c. A DHCP-szerver a következő adatokat küldheti a számítógépnek
        i. IP-cím: A számítógép egyedi címe a hálózaton.
        ii. Alhálózati maszk: Az alhálózat azonosítására.
        iii. Alapértelmezett átjáró (default gateway): Az internetelléréshez szükséges útvonal.
        iv. DNS-szerver címe(i): Az a kiszolgáló vagy kiszolgálók IP-címe, amelyeket a számítógép használhat a domainnevek IP-címekre való feloldására
    d. A számítógép automatikusan elfogadja a kapott DNS-szerver címét, és azt beállítja a hálózati adapteren keresztül
12. Mit jelent az iteratív címfeloldás?
    a. Az iteratív címfeloldás a DNS (Domain Name System) működésében azt jelenti, hogy a DNS-kliens (általában a számítógép vagy más eszköz) a DNS-kérést sorozatosan küldi el különböző DNS-szervereknek, hogy megtalálja a keresett domain névhez tartozó IP-címet
13. Magyarázza el az iteratív címfeloldás működését az index.hu cím esetén.
14. Milyen protokollokat alkalmazunk az elektronikus levelezés során? Mi az egyes protokollok feladata?
    a. SMTP
        i. Simple Mail Transfer Protocol
        ii. Feladata
            - Levélfeladás (a szolgáltató levelező szerverére)
            - Levéltovábbítás (a címzett levelező szerverére)
        iii. ASCII protokoll
        iv. Manapság már biztonságos változatait használjuk (TLS felett)

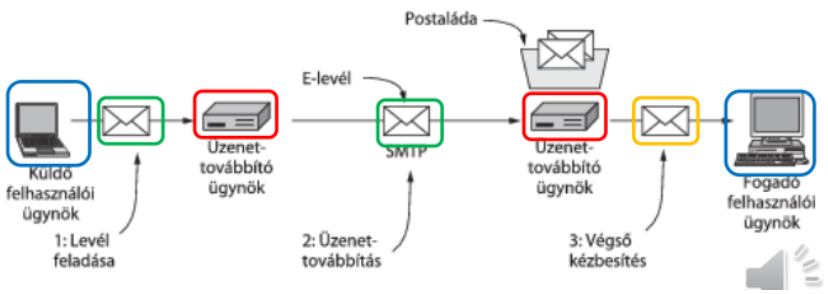


b. IMAP
   i. Internet Message Access Protocol
   ii. POP3 továbbfejlesztett változata
   iii. Feladata:
      - Postaláda kezelése
      - Levelek letöltése

- Fontosabb parancsok
  - LOGIN *név jelszó*
  - LIST "" "*"
  - SELECT *foci*
  - STATUS *foci* (MESSAGES)
  - FETCH *sorszám* (BODY[HEADER])
  - FETCH *sorszám* (BODY[szám])
  - LOGOUT

c. MIME
   i. A kezdeti levéltovábbító protokollok
      - …ASCII karaktereket vártak
      - Egy sor nem lehetett 1000 karakternél hosszabb
   ii. De szeretnénk mást is továbbítani (képek, kód, video, …)
   iii. Megoldás: MIME (Multipurpose Internet Mail Extensions)
   iv. Gyakori kódolás: base64
      - Kód 6 bites csoportokra bontása
         a. 3 bájt→4 db 6-bites csoport
      - 64 karakterrel kódoljuk:
         a. A-Z, a-z, 0-9, +, /
      - == és =
         a. jelzi, ha „nincs tele” az utolsó csoport

15. Ismertesse az elektronikus levelező rendszerek felépítését, működését.
   a. Felhasználói ügynök
      i. Üzenetek olvasása, küldése (feladása)
      ii. SMTP-t használ
   b. Üzenettovábbító ügynök
      i. (levelező szerver)
      ii. Levelek eljuttatása a feladótól a címzett postaládájába
      iii. POP3, IMAP használ

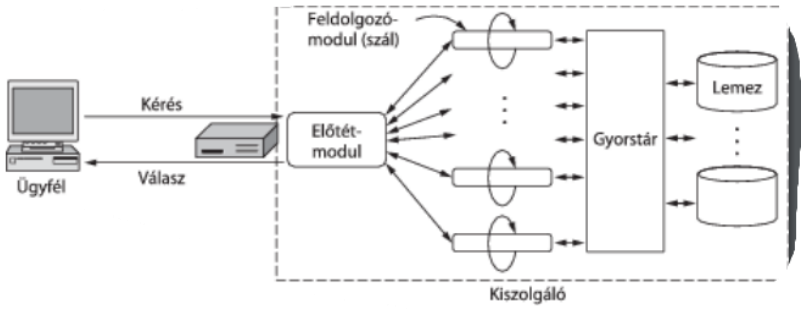


16. Mi az URL? Ismertesse az URL felépítését.
    a. Universal Resource Locator
    b. Pl.: http://uni-obuda.hu/oktatas
        i. Protokoll (vagy séma): http
        ii. Hoszt DNS neve: uni-obuda.hu
        iii. Út: oktatas
17. Ismertesse a legfontosabb URL-sémákat (protokollokat), ezek célját.

Név	Rendeltetés	Példa
http	Hipertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hipertext biztonságos átvitele	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Helyi állomány	file:///usr/suzanne/prog.c
mailto	E-levél küldése	mailto:JohnUser@acm.org
rtsp	Valós idejű média letöltése	rtsp://youtube.com/montypython.mpg
sip	Multimédiás hívások	sip:eve@adversary.com
about	Böngésző információ megjelenítése	about:plugins

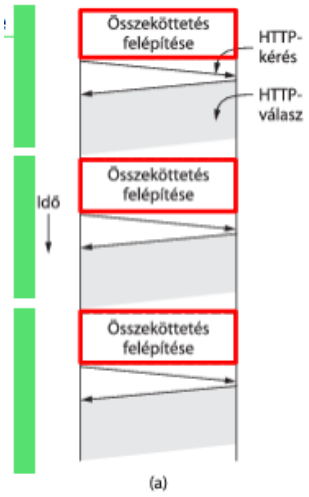
18. Ismertesse egy web-oldal elérése során a kliens oldal működését.
    a. A böngésző meghatározza az URL-t (megnézi, hogy mit választottak ki)
    b. A böngésző megkérdezi a DNS-től a szerver IP-címét.
    c. A DNS válasza: 128.208.3.88.
    d. A böngésző létrehoz egy TCP-összeköttetést a 128.208.3.88 című hoszt 80-as portjával, a HTTP-protokoll jól ismert portjával.
    e. Átküld egy kérést, melyben elkéri az /index.html oldalt
    f. A kiszolgáló elküldi az oldalt egy HTTP-válasz formájában: átküldi az /index.html állományt.
    g. A böngésző ugyanezzel a módszerrel lekéri az oldal többi URL-jét is.
        i. css fájl, beágyazott kép, beágyazott videó, parancsfájl, stb.
    h. A böngésző megjeleníti az /index.html oldalt.
    i. A TCP-összeköttetést lebontják.
19. Ismertesse egy web-oldal elérése során a szerver oldal működését.
    a. A szerver elfogad egy TCP-összeköttetést az ügyféltől (a böngészőtől).
    b. Megkapja az oldalhoz vezető utat, ami a kért állomány neve (/index.html).
    c. Megkeresi az állományt (a háttértáron)
    d. Elküldi az állomány tartalmát az ügyfélnek.
    e. Bontja a TCP-összeköttetést.

20. Ismertesse a modern web-szerverek vázlatos felépítését. Mi a front-end és back-end feladata?



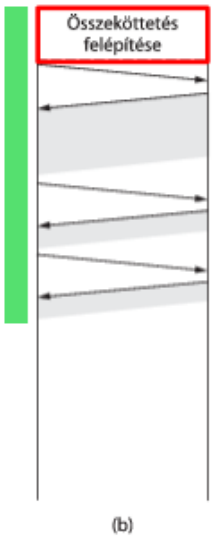
    a. Előtét-modul (front-end)
        i. Fogadja a kéréseket
        ii. Átadja a kérést az egyik feldolgozó szálnak
    b. Feldolgozó modulok (külön szálakon)
        i. Ellenőrzi a gyorstárat (ha van találat, akkor azt használja)
        ii. Ha nincs találat, előveszi az oldalt a háttértárról
        iii. Gyorstárba beírja
        iv. Eredményt visszaküldi az ügyfélnek
21. Mit jelent, hogy egy web-oldal statikus vagy dinamikus?
    a. Statikus:
        i. A weboldal tárolt formáját vesszük elő (pl. előző oldali szerver)
        ii. Minden híváskor ugyanúgy néz ki.
        iii. Pl.: logók, stíluslapok
    b. Dinamikus:
        i. A kérés feldolgozása után állítjuk elő az oldalt
        ii. Az oldal dinamikusan változhat is
22. Ismertesse egy dinamikus web-oldal letöltésének menetét. Térjen ki a kliens-oldalon futó programok (pl. java-script) szerepére is.
        i. Kérés
        ii. Szerver-oldali program előállítja a választ
        iii. Válasz visszaküldése
        iv. Kliens-oldali program aktiválása
        v. Újabb kérés
        vi. Szerver-oldali program futása…
        vii. Válasz, oldal frissül
23. Mi a HTTP és a HTTPS?
    a. HTTP: HyperText Transfer Protocol
        i. Kérés-válasz protokoll TCP felett
            - Egy TCP kapcsolat - sok http kérés is mehet rajta
                a. Tartós kapcsolat – persistent connections (HTTP1.1 óta)
        ii. Metódusokat használ
            - Kérések és válaszok fejlécei: ASCII
            - Tartalom: MIME formátumokban
    b. HTTPS: A HTTP biztonságos változata
        i. Működése hasonló, de titkosított csatornán történik a kommunikáció

24. Ismertesse a http kérések és válaszok működését a HTTP 1.0 alatt.
    a. Minden kérés külön TCP összeköttetésen



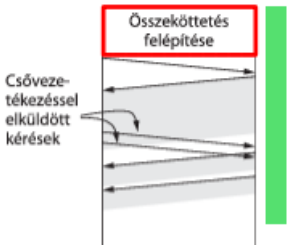
1. ábra: Zölddel a TCP kapcsolat ideje

25. Ismertesse a http kérések és válaszok működését a HTTP 1.1 alatt. Mivel magyarázható a gyorsulás a HTTP 1.0-hoz képest (2 ok).



    a. A HTTP 1.1 lehetővé teszi, hogy egy TCP-kapcsolatot többször felhasználjanak, tehát egyetlen kapcsolatot több kérés és válasz kezelésére tartanak fenn.
    b. A kérés (request) tartalmazza:
        i. HTTP metódusokat, pl. GET, POST.
            - A kért erőforrás URL-jét.
            - Fejlécadatokat (például milyen típusú válaszokat vár a kliens).
    c. A szerver válasza (response)
        i. Egy állapotkóddal (pl. 200 OK, 404 Not Found).
        ii. A válasz tartalmával (pl. egy weboldal HTML kódja).
        iii. Fejlécadatokkal (például a válasz típusát és hosszát).
    d. Gyorsulás okai
        i. Az összes kérés egy TCP összeköttetésen
        ii. HTTP 1.1 lehetővé teszi, hogy az ügyfél több kérést küldjön egyetlen TCP-kapcsolaton belül, anélkül, hogy megvárná az előző kérés válaszát

26. Ismertesse a http kérések és válaszok csővezetékes működését a HTTP 2 alatt. Mivel magyarázható a gyorsulás a HTTP 1.1-hoz képest?



    a. Csővezeték használata
    b. A HTTP/2 nagy újítása a bináris protokoll használata, ami gyorsabb és hatékonyabb, mint a HTTP/1.1 szöveges protokollja
    c. Egyetlen TCP-kapcsolaton belül több kérés és válasz egyszerre, párhuzamosan haladhat, anélkül, hogy várakozniuk kellene egymásra
    d. A kérések és válaszok kisebb bináris keretekre osztódnak, amelyek függetlenül érkezhetnek és állhatnak össze az ügyfél vagy a szerver oldalán

27. Sorolja fel a fő HTTP metódusokat és azok célját.

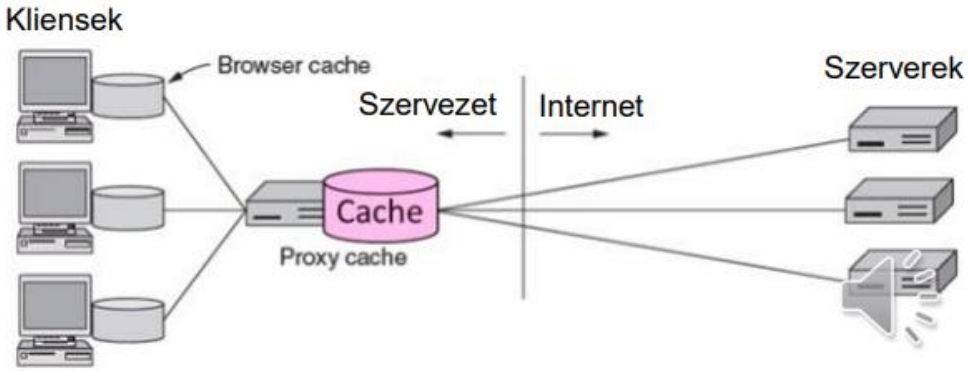
Metódus	Leírása
GET	Weboldal olvasása
HEAD	Weboldal fejlécének olvasása
POST	Weboldalhoz történő hozzáfűzés
PUT	Weboldal tárolása
DELETE	Weboldal eltávolítása
TRACE	Bejövő kérés visszaküldése
CONNECT	Kapcsolódás proxy-n keresztül
OPTIONS	Egy oldal opcióinak lekérdezése

28. Soroljon fel néhány ismert állapotkódot.

Kód	Jelentése	Példák
1xx	Információ	100 = a kiszolgáló jóváhagyja az ügyfél kérését
2xx	Siker	200 = sikeres kérés; 204 = nincs tartalom
3xx	Átirányítás	301 = az oldal elköltözött; 304 = a gyorstárban tárolt oldal még érvényes
4xx	Ügyfél hibája	403 = tiltott oldal; 404 = az oldal nem található
5xx	Kiszolgáló hibája	500 = belső hiba a kiszolgálóban; 503 = próbálkozzon újra később

29. Hogyan működik a böngészőbe épített gyorstár?
    a. Letöltött oldalakat a gyorsító tárban tároljuk
    b. Ha újra hivatkozunk rá, akkor a gyorstárból töltjük be
30. Hogyan lehet oldal érvényességét ellenőrizni a gyorstárban?
    a. meddig érvényes egy tárolt oldal?
    b. „Lejár” fejléc: ha van, akkor ez mutatja az érvényességi időt
    c. Feltételes GET
        i. Ha az oldal nem változott, akkor rövid válasz, különben a teljes oldal
            - ha nem változott nem rendelődik hozzá erőforrás (nem terhelődik a hálózat)
            - ha változott akkor a teljes oldal tartalma a gyorstárba kerül

31. Ismertesse a proxy szerverek működését, fő feladatait. Mik ezen megoldás korlátai?
    a. Szervezetek használhatnak saját proxy szervert
        i. ISP, cég, egyetem
        ii. Tárolja a szervezet bármely felhasználója által letöltött oldalakat a proxy cache-ben
        iii. Ha bárki újra hivatkozik rá, akkor a gyorstárból töltjük be
    b. Egyéb feladatokat is ellát
        i. Biztonság (tűzfal)
        ii. Tartalomszűrés
    c. Korlátok:
        i. Népszerűtlen oldalak
            - Ha az oldal nem népszerű, akkor valószínűleg nem lesz a proxy cache-ben tárolva, ezért nem érhető el gyorsabban, mint közvetlen lekérdezés esetén
        ii. https!!!
            - a proxy nem férhet hozzá a titkosított adatfolyamhoz, hacsak nem használják a "man-in-the-middle" (MITM) proxy technikát

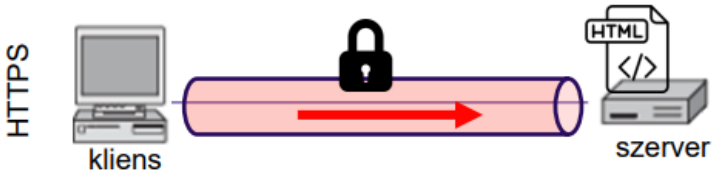


32. Hasonlítsa össze a HTTP1.1, HTTP2 és a HTTP3 főbb tulajdonságait. Ismertesse a hatékonyság növelése érdekében bevezetett új megoldásokat.

Protokoll	Bevezetés éve	Hálózati réteg	Fő jellemzők
HTTP/1.1	1997	TCP felett	- TCP felett működik
HTTP/2	2015	TCP felett	- Kérések csővezetéken - Tömörített fejrész - Szerver push (nem kért információt is küld)
HTTP/3	2020 (Internet Draft)	QUIC felett	- QUIC felett működik - A legtöbb böngésző már támogatja

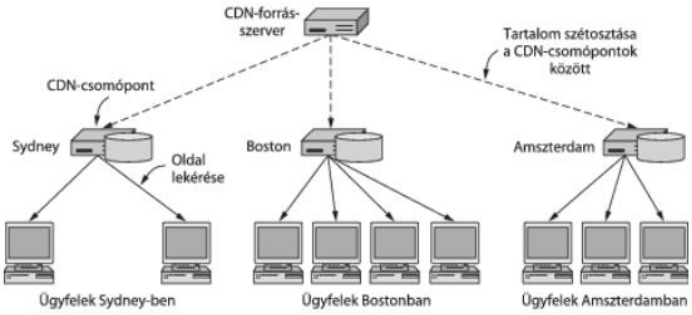
33. Miért előnyösebb a HTTPS protokoll használata a HTTP-vel szemben?
    a. Problémák a HTTP-vel:
        i. Lehallgatás
        ii. Üzenetek módosítása
        iii. Megszemélyesítés

34. Ismertesse a HTTPS vázlatos működését. Mire szolgál a kétkulcsos titkosítás? Mire szolgálnak a tanúsítványok?



   a. Kétkulcsos titkosítás (üzenetváltás biztonságos)
      i. Nyilvános kulcsot a szerver elküldi
   b. A nyilvános kulcs valódiságát a böngésző ellenőrzi
      i. Tanúsítvány (Certificate)

35. Miért van szükség speciális tartalomszállító hálózatokra?
   a. CDN a rövidítése
   b. Ötlet:
      i. A népszerű tartalmakat a fogyasztó közelébe kell vinni
      ii. Azonos tartalom több helyen (több példányban)
      iii. Az optimális kiszolgáló példánytól (pl. a legközelebbitől) kapjuk az adatot
   c. A tartalom szétosztása CDN csomópontokba
   d. A kérések kiszolgálása a megfelelő csomópontban



   e. Kérések kezelése tartalomszállító hálózatokban
      i. A tartalomszolgáltató saját, módosított DNS szervereket üzemeltet
      ii. A DNS szerver a kérésből tudja, honnan érkezett
      iii. Ennek megfelelő választ küld vissza

36. Ismertesse a szerverfarmok működését.
   a. Egyetlen logikai webhely
   b. Sok kiszolgáló gép
   c. Melyik gép szolgálja ki az aktuális kérést?
      i. Előtétberendezés (front-end)
      ii. Szétszórja a kéréseket a szerverfarm kiszolgálói között
      iii. Egy web-kérés minden csomagját ugyanazon kiszolgálóhoz kell irányítani!
      iv. Módszerek:
      v. Front-end mindenkihez továbbít, a szerver maga szűr
      vi. pl. IP-cím alapján
      vii. Front-end szűr és optimalizál
      viii. Terheléskiegyenlítés (load balancing)

Mire használjuk az RTP protokollt?

Szerver több forrást is használhat (pl. video, audio különféle nyelveken)
Adatokat RTP blokkokba kódolják / RTP blokkokat UDP-n keresztül továbbítják

Mire használjuk az RTCP protokollt?

Visszacsatolást kezeli: Hálózat tulajdonságairól a szervernek (késleltetés, jitter, sávszélesség, torlódás) Ez alapján képes folyamatosan állítani a minőséget (pl. kódolás, felbontás)
Szinkronizációt biztosít:. A különféle folyamok óráit egymáshoz szinkronizálja
Forrás-információkat szállít Pl. a beszélő neve

Milyen hatása van a késleltetésnek és a jitternek a valós idejű protokollok működése során?

Késleltetés: Sok alkalmazásban nincs lényegi hatása (pl. zenehallgatás) / Egyes alkalmazásokban kis értéken kell tartani (pl. telefon)
Késleltetés ingadozása (jitter): Nagyon zavaró! Kompenzálni kell a vevőben!

Hogyan kompenzáljuk a jitter hatását a vevőben? Mit jelent a lejátszási pont?

-Pufferelés a vevőben és Lejátszási pont (playback point):
– Mennyit kell várnia vevőnek a lejátszás indításáig
– Pl. a csomagok 99%-a megérkezzen • Lejátszási pont változtatása: – Pl. beszédszünetben

12. Hogyan azonosítunk egy csatlakozót (címzés)?

Küldő és fogadó létrehoz egy csatlakozót (socket)
b.) A socket címe három elemből áll:
IP cím (pl. IPv4 vagy IPv6) <- NSAP // Port száma (16 bites egész) <- TSAP / Protokoll (pl. TCP, UDP)

13. Mit jelent a „jól ismert port”? + Magyarázza el az inted szerepét és működését.

a.) A szerverek „Ismert” portokat használnak („Well Known Ports”)
1024 alatti portok // Csak privilegizált felhasználóknak // Minden porthoz egy démon tartozik
Gyakran egyetlen felügyelő démont használunk (pl. inetd – Internet Daemon)
inetd figyel több portot is // szükség esetén elindítja a megfelelő démont

14. Milyen portszámokat kapnak a kliens folyamatok? Kitől kapják a portszámokat?

a.) Általában ideiglenes portokra csatlakoznak
b.) Az operációs rendszer választja ami Véletlenszerű

Mit jelent, hogy a TCP full-duplex és kétpontos kommunikációt biztosít?

A TCP full-duplex: az adatküldés és fogadás egy időben, kétirányúan történik mindkét kommunikáló fél között.
A kétpontos kommunikáció TCP egy adott kapcsolat mindig két konkrét végpont (pl. kliens és szerver) között zajlik, nincs adatszórás (broadcast) vagy többesküldés (multicast).
Ezek biztosítják a megbízható, kétirányú adatátvitelt egyetlen kapcsolat keretein belül.

Mit jelent, hogy a TCP egy bájtfolyamat biztosít?

az adatokat folyamatos bájtok sorozataként kezeli, nem pedig különálló üzenetekként. Az alkalmazás által küldött adatok sorrendje megmarad, de nincs üzenethatár. A TCP feladata a szegmensek összeállítása és sorrendhelyes kézbesítése.

Mikor küldi el a TCP a rábízott adatokat? Hogyan lehet gyors továbbítást kikényszeríteni?

Az elküldendő adatot szabad belátása szerinti időben küldi
Lehet pufferelni, hogy nagyobb szegmens összegyűljön
PUSH bit: kérheti a TCP-t, hogy ne késleltessen (pl. online játék)

A TCP milyen adategységet sorszámoz?

Minden bájt rendelkezik egy 32 bites sorszámmal Ezeket a kapcsolat kezdetén meghatározott kezdeti sorszámhoz viszonyítva kezeli

Biztosítja a bájtfolyam sorrendiségét, a hiányzó csomagok felismerését és az újraküldési mechanizmus működését

Mi korlátozza TCP szegmens méretét?

IP adatmező -> MTU (legnagyobb átvihető adategység) -> Pl. Ethernet: 1500B

Magyarázza el a TCP fejrészben a sorszám és a nyugtaszám szerepét.

Sorszám (SEQ): A korábban elküldött bájtok száma

Nyugtaszám (ACK): Halmozott nyugta – Jelzi, hogy a nyugtázott bájtig az összes adat hiánytalanul megérkezett – A nyugta valójában a várt bájt indexét tartalmazza (rendben vett bájt sorszáma + 1)

21. Magyarázza el a TCP fejrészben az ablakméret mező szerepét.

A vevőben rendelkezésre álló puffer mérete

22. Magyarázza el a TCP fejrészben a CWR és ECE bitek szerepét.

Vevő ECE: bittel jelez: torlódás van a hálózaton

Adó CWR: bittel jelez vissza, hogy a forgalmat (a torlódási ablaka méretét) lecsökkentette

Magyarázza el a TCP fejrészben az ACK, PSH, RST, SYN és FIN bitek szerepét.

a.) **ACK:** Jelzi a nyugtaszám érvényességét

b.) **PSH (PUSH):** késedelem nélküli továbbítás kérése i. Ne legyen pufferelés

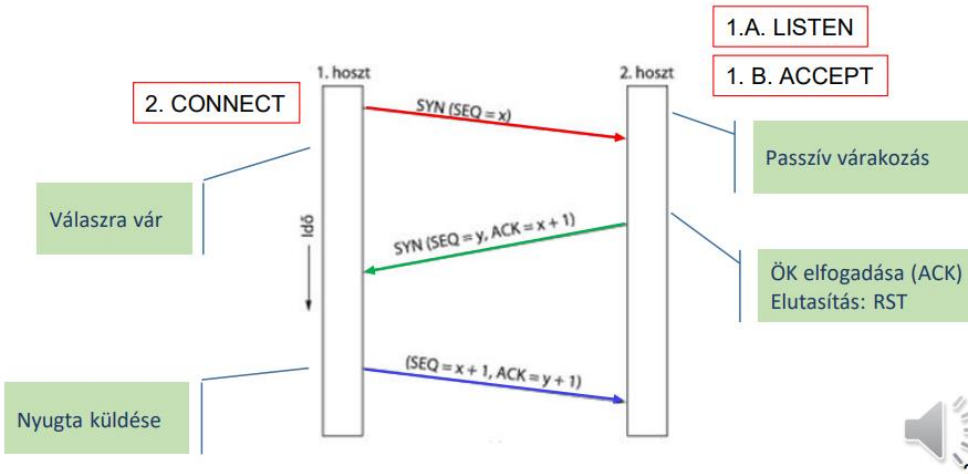
c.) **RST (RESET):** Összeköttetés helyreállítás (valami baj történt)

d.) **SYN:** kapcsolat kiépítés ->Jelzi a CONNECTION REQUEST és CONNECTION ACCEPTED üzeneteket

e.) **FIN:** kapcsolat bontása i. Jelzi, hogy a küldőnek nincs több adata

Ismertesse, hogyan alkalmazza a TCP az összeköttetés létrehozásához a háromutas kézfogást

- →SYN=1, ACK=0
- ←SYN=1, ACK=1
- →SYN=0, ACK=1

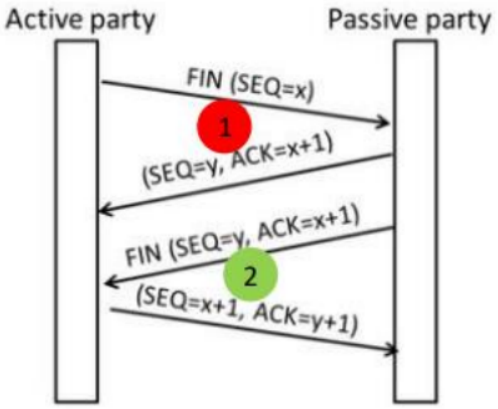


Ismertesse a TCP összeköttetés lebontásának módját.

Időzítők használata -> Ha FIN-re nem jön válasz, bontja az ÖK-t

i. FIN jelzi, hogy nem kíván több adatot küldeni

→FIN: részemről nincs több adat
←ACK: OK, vettem
←FIN: Részemről sincs több adat
→ACK: OK, vettem



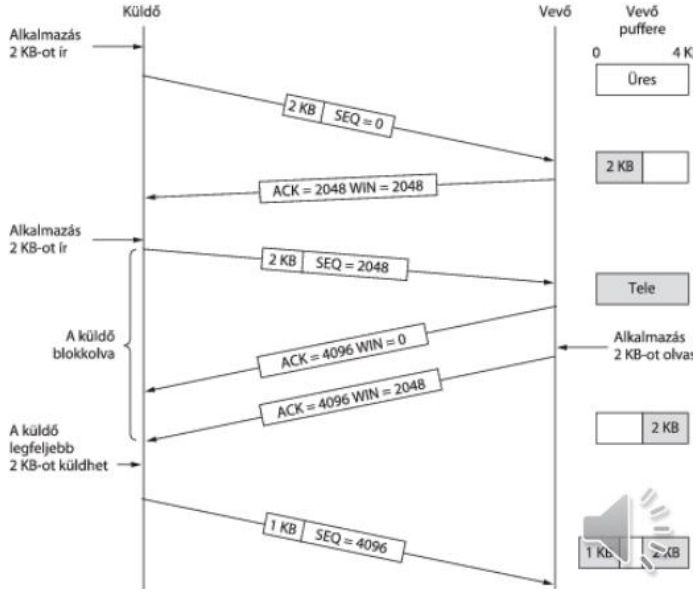
Magyarázza el a TCP csúszóablakos forgalomszabályozásának működését

Különválik az ACK és a vevő pufferméretének kezelése

ACK: halmozott nyugta // A nyugtázott bájtig az összes adat hiánytalanul megérkezett (ACK a következő – várt – sorszámot tartalmazza)

WIN: vevőben rendelkezésre álló ablakméret i. = maximális küldhető adatmennyiség

SEQ: sorszám -> Az eddig elküldött bájtok száma (ez a csomag nem számít bele)



Hány bájtot küldhet a kliens a szerver felé, ha a szerver utolsó üzenete a következő volt:

ACK=67123, WIN=512 // ACK=512, WIN=0 Mivel a WIN=512, szerver rendelkezésre **álló ablakmérete 512**, tehát ennyi adatot küldhet még max a kliens. A kliens küdlött egy ACK-t válaszul tehát még 511 hely van vissza.

29. Egy szervernek 2kB üres pufferterület áll rendelkezésre egy TCP összeköttetés kezelésére. Most a következő (512 adatbájtot tartalmazó) üzenet érkezik a szerverhez: ADAT: 512B, SEQ=3000 ***ACK=3512 WIN=1536***, mert jelzi, hogy fogadta az 512B adatot, mivel a 3000-ik sorszámként küldte így a következő, ami jön az **a 3000+512=3512**-ik sorszám lesz. Jelzi azt is, hogy **2kB (2048B)** a puffere mérete, amiből ez az 512B lejön szal **2048-512=1536.**

Hogyan lehet a szegmens elvesztését időzítő segítségével detektálni? Hogyan célszerű beállítani az időzítő értékét, ha ismerjük az átlagos késleltetési időt (SRTT) és a késleltetési idő szórását

Ha a szegmensre **nem jön az időzítő lejárta előtt válasz**, akkor **újraküldjük**

Az időzítő, ha 500ms akkor túl kicsi, ha 900ms akkor túl nagy

Változtassuk az időzítő értékét az aktuális helyzetnek megfelelően - adaptív időzítő • átlag és • szórás kell hozzá

Hogyan becsüljük a késleltetési idő átlagos értékét? Ism.. az exponenciális átlagoló működését.

Exponenciális átlagolással: az aktuálisan mért értékeket és a korábbi simított értékeket kombinálja egy adott súlyozási tényezővel **RTT:** mért körülfordulási idő (Round Trip Time)

SRTT: simított körülfordulási idő (Smoothed Round Trip Time) **sVAR:** az RTT simított „varianciája”

$$SRTT_{\text{új}} = \alpha \cdot SRTT_{\text{régi}} + (1-\alpha)RTT$$

$$sVAR_{\text{új}} = \beta \cdot sVAR_{\text{régi}} + (1-\beta)\left|SRTT_{\text{új}} - RTT\right|$$

$\alpha = 7/8$
$\beta = 3/4$

Alfa és béta súlyozások, a kisebb értékek kisebb, a nagyobb értékek súllyal vannak figyelembevéve

Ism a „3 ismételt nyugta” szabályt. Hogyan lehet ennek segítségével a szegmens elveszését detektálni?

A szegmens valószínűleg elveszett, ha nem jött rá nyugta RTO időn belül • Ehhez ki kell várni az RTO időt • Lehetne gyorsabban is detektálni az elveszett csomagot? Igen... 3 nyugtamásolat szabály legalább 3 ismételt nyugta (nyugtamásolat) jön egymás után

Mitől jöhet létre torlódás? Mit jelent a torlódási ablak? Hogyan használja a TCP a forgalomszabályozási és a torlódási ablakot?

Ha a hálózati terhelés túl nagy Csomagok feltorlódnak az útválasztókban Csomagok késnek, elvesznek **TCP:** torlódási ablakot használ -> Mennyi adat lehet a hálózaton egyszerre? Hasonló a forgalomszabályozási ablakhoz • A TCP együtt használja a két ablakot • Amelyik kisebb, annak megfelelő mennyiségű adatot küld ki

Ismertesse a nyugtaórajel működését

Egy hálózat sebességét a leglassabb szakasza határozza meg

Egy gyors löketre válaszul visszaérkező nyugták sebessége azt mutatja meg, hogy milyen gyorsan lehet a csomagokat a hálózat leglassabb részén átvinni

adó ennek megfelelő sebességgel ad -> Elkerüli a felesleges sorbanállást az útvonalválasztókon „sima” kimenőforgalmat biztosít

Ismertesse a Lassú kezdés algoritmus működését. Mi célt szolgál ez az algoritmus a TCP-ben?

torlódási ablakot beállítsa: Ha nincs torlódás, növeljük (AI) Ha torlódás van, csökkentsük (MD)

A kezdeti munkapont elérése AI-val lassú indításkor exponenciálisan növekvő ablakméretet

Ez lesz a „lassú kezdés” algoritmus (Slow Start) • Minden nyugta vételénél ablakméret eggyel nő // új csomagot küld a régi helyett // új csomagot küld az új üres helyre

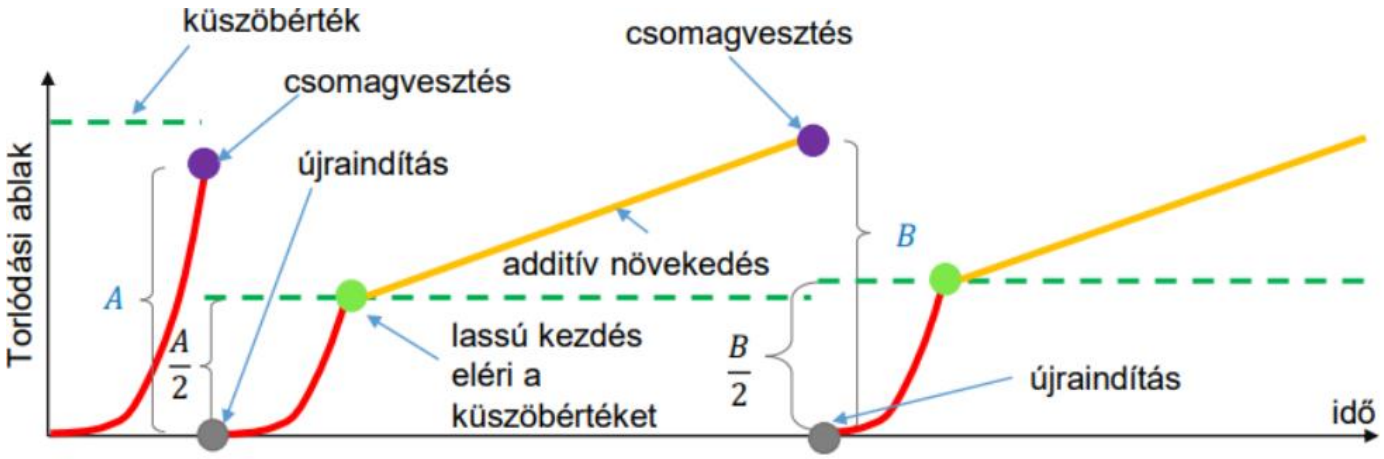
• Úton lévő csomagok száma RTT alatt duplázódik

Mit értünk „gyors újraküldésen” a TCP Tahoe protokollban?

Torlódás/csomagvesztés érzékelése Időtúllépés vagy 3 nyugtamásolat

• ekkor a hiányzó csomagokat azonnal újraküldjük • Ez a gyors újraküldés (fast retransmission)

Ismertesse a TCP Tahoe torlódáskezelését a lassú kezdés segítségével. Használja a 36. oldal ábráját.



Soroljon fel alkalmazási rétegben megvalósított protokollokat. Ismertesse ezek feladatát is

DNS - körzetnevek IP-címekké alakítása **SMTP**:e-mail küldés
POP3:Post Office Protocol e-mail letöltése a szerverről **IMAP:**e-mail elérése a szerveren
FTP: Megbízható fájlátvitel két gép között **http/S,** WWW használata **BitTorrent** P2P fájlmegosztás

Milyen szolgáltatásokat használnak az alkalmazási réteg protokolljai?
A szállítási réteg szolgáltatásait használja pl HTTP, FTP, DNS -> **TCP UDP használ**

Mi a DNS, mi a célja? névfeloldó rendszer, amely ember által olvasható domainneveket (pl. example.com) fordít le gépek által értelmezhető IP-címekre (pl. 192.0.2.1).

Mi volt a hosts.txt fájl feladata a korai hálózatokban? Mi hosts fájl feladata mai rendszerben
ARPANET hosts.txt fájl -> összes számítógép nevét IP-címét, és minden gép éjszakánként frissítette ezt a fájlt.
internet növekedett kezelhetetlenné ->1983 bevezették DNS hierarchikus/elosztott névkiosztást biztosított.
A mai helyi névfeloldásra, például egy gép gazdanevének és IP-címének manuális összerendelésére.

Ismertesse a DNS működését. Mi a resolver csonk feladata? Mi a helyei DNS szerver feladata?
DNS domainnevek IP-címekké való fordítását végzi. **resolver csonk** felhasználói programból indít kérést, amelyet a helyi DNS szervernek továbbít. A helyi DNS szerver iteratív kereséssel oldja fel a nevet más DNS szerverek segítségével. Az IP-címet visszaküldi a resolver csonknak, amely továbbítja az eredményt a programnak, így az kommunikálhat a keresett szerverrel.

Mi az ICANN, mi a feladata?
internet cím és névterének működését koordinálja. Feladata DNS, IP-címek, a protokoll-azonosítók és a DNS gyökérzóna kezelése, biztosítva az internet globális és zökkenőmentes működését.

Ismertesse a DNS névtér hierarchikus felépítését.
DNS névtér hierarchikus felépítésű, pontokkal elválasztott körzetekkel.Az elsődleges körzeteket (.com, .hu) ICANN kezeli, és több mint 1000 létezik, általános és országkódos tartományokkal.
A másodlagos körzeteket (pl. cisco.com, uni-obuda.hu) regisztrátorok adminisztrálják.
Minden körzet maga felügyeli az alatta lévő körzetek kiosztását, pl. amk.uni-obuda.hu. A névfeloldás balról jobbra történik, haladva a gyökér felé.

Mit tartalmaz egy DNS bejegyzés (5)? Körzet-név / Élettartam érvényességi idő másodpercben
Osztály általában IN (internet) **Típus** az adat típusa **Érték** típusnak megfelelő érték

Mit jelent a hiteles és tárban lévő bejegyzés?
Hiteles bejegyzés: Az információ a bejegyzést kezelő szervtől származik (biztosan helyes)
Tárban lévő bejegyzés: A bejegyzés a gyorstárból való (esetleg idejétmúlt lehet)

Automatikus DNS cím beállítása esetén hogyan kapja meg a számítógép a DNS szerver címét?
DNS-szerver címét a DHCP segítségével kapja meg. A számítógép **DHCP Discover** üzenetet küld, a DHCP-szerver pedig válaszként elküldi az IP-címet, alhálózati maszkot, alapértelmezett átjárót és a DNS-szerver címét, amelyet a gép automatikusan beállít.

Mit jelent az iteratív címfeloldás? -> DNS-kliens több DNS-szerverhez küld kéréseket egymás után, amíg meg nem találja a keresett domainnévhez tartozó IP-címet.

Milyen protokollokat alkalmazunk az elektronikus levelezés során? Mi az egyes protokollok feladata?
SMTP: Feladata a levél feladása és továbbítása a címzett levelező szerverére.
ASCII protokoll alapú, manapság biztonságos változatait használjuk (TLS felett)
IMAP: POP3 továbbfejlesztett változata Feladata: Postaláda kezelése Levelek letöltése
MIME: kezdeti levéltovábbító protokollok ASCII karaktereket vártak 1sor nem lehetett 1000 karakternél hosszabb

Ismertesse az elektronikus levelező rendszerek felépítését, működését.
Felhasználói ügynök: Az üzenetek olvasására és küldésére szolgál. SMTP protokollt használja.
Üzenettovábbító ügynök: Felelős a levelek eljuttatásáért a feladótól a címzett postaládájába. Ehhez az SMTP-t használja a feladáskor, míg a levelek letöltésére és olvasására POP3 vagy IMAP

Mi az URL? Ismertesse az URL felépítését.
URL: internetes erőforrások elérési útját jelöli. Felépítése a következő elemekből áll:
Protokoll (séma): http // **Hoszt** DNS neve: uni-obuda.hu // **Út:** /oktatas

Ismertesse a legfontosabb URL-sémákat (protokollokat)
http: Hypertext Weboldalak elérésére használt protokoll. pl hhtp://miutdomén.com
https: Hypertext Secure - Biztonságos HTTPS kapcsolat, titkosítással. pl hhtpS://miutdomén.com
ftp: File Transfer Protocol - Fájlok átvitelére szolgáló protokoll.
file: Helyi fájlrendszeren található fájlok elérése.
mailto: E-mail küldésére szolgáló protokoll.
rtsp: Real-Time Streaming Protocol - Valós idejű médiafolyamok lejátszására.
sip: Session Initiation Protocol - Hívások kezdeményezésére használják, például VoIP hívásokhoz.
about: Böngésző specifikus információk megjelenítése.

Ismertesse egy web-oldal elérése során a kliens oldal működését.
A böngésző **meghatározza az URL-t.**
A böngésző a **DNS-től lekéri a szerver IP-címét** (pl. 128.208.3.88).
A böngésző **TCP-összeköttetést hoz létre a 80-as porton a szerverrel.**
A böngésző HTTP kérést küld az /index.html oldal lekérésére.
A kiszolgáló HTTP-választ küld, és átadja az /index.html fájlt.
A böngésző **további URL-eket is lekér (pl. CSS, képek, videók).**
A böngésző **megjeleníti az /index.html oldalt**.
A TCP-összeköttetést lebontják.

Ismertesse egy web-oldal elérése során a szerver oldal működését.

A szerver elfogadja a TCP-összeköttetést az ügyféltől (böngészőtől).
Megkapja a kért állomány elérési útját (pl. /index.html).
Megkeresi az állományt a háttértáron. Elküldi az állomány tartalmát az ügyfélnek. Lebontja a TCP-összeköttetést.

Ismertesse a modern web-szerverek vázlatos felépítését. Mi a front-end és back-end feladata?

Előtét-modul (front-end): Fogadja a kéréseket. / Átadja a kérést a feldolgozó szálaknak.
Feldolgozó modulok (back-end): Ellenőrzik a gyorstárat, és ha találat van, azt használják.
Ha nincs találat, a háttértárról előveszik az oldalt. A gyorstárba írják az eredményt. Visszaküldik az eredményt az ügyfélnek.

Mit jelent, hogy egy web-oldal statikus vagy dinamikus?

Statikus: Az oldal tárolt formáját jeleníti meg minden híváskor ugyanúgy, például logók vagy stíluslapok esetén.
Dinamikus: Az oldal a kérés feldolgozása után jön létre, és a tartalma változhat a felhasználó vagy más környezeti tényezők alapján.

Ismertesse egy dinamikus web-oldal letöltésének menetét. Térjen ki a kliens-oldalon futó programok (pl. java-script) szerepére is.

A kliens kérés küld a szervernek. -> A szerver-oldali program előállítja a választ.
A válasz visszaküldése a kliensnek. -> A kliens-oldali program (pl. JavaScript) aktiválódik.
A kliens újabb kérést küldhet, ha szükséges. -> A szerver-oldali program fut, és feldolgozza a kérdést.
Válasz érkezik, és az oldal frissül a kliens oldalon.

Mi a HTTP és a HTTPS?

http: kérés-válasz alapú protokoll, amely a TCP kapcsolat felett működik, és lehetővé teszi a weboldalak elérését. A HTTP 1.1 óta támogatja a tartós kapcsolatokat, így egy TCP kapcsolaton több kérés is lebonyolítható. A kommunikáció ASCII formátumban történik, és MIME formátumokat használ a tartalomhoz.
HTTPS: HTTP biztonságos változata, amely titkosított csatornán keresztül biztosítja a kommunikációt, így megvédi az adatokat az illetéktelen hozzáféréstől. 1/1 -> HTTP, de SSL/TLS titkosítást alkalmaz.

HTTP 1.0 Kérések és Válaszok Működése

a kommunikáció egy kérés-válasz párban zajlik. Minden egyes kéréshez új TCP kapcsolatot kell létrehozni, amelyet a szerver válasza után bezárnak. Ez azt jelenti, hogy minden egyes új kérés új kapcsolatot igényel, ami lassítja a weboldalak betöltését.

2. HTTP 1.1 Kérések és Válaszok Működése és előnye

Tartós kapcsolat: A TCP kapcsolatot több kérés és válasz kezelésére tartják fenn, így nem szükséges minden új kéréshez új kapcsolatot létrehozni, ami gyorsítja a kommunikációt.
Pipelining: A HTTP 1.1 lehetővé teszi, hogy az ügyfél több kérést küldjön egyetlen TCP-kapcsolaton belül, anélkül, hogy megvárná az előző kérés válaszát. Ez jelentősen csökkenti a késleltetést.

3. HTTP 2 Csővezetékes Működés és Gyorsulás + újfunciók

Bináris protokoll: A HTTP 2 szöveges helyett bináris protokollt használ, amely gyorsabb és hatékonyabb, mivel könnyebben feldolgozható.
Multiplexálás: Több kérés és válasz párhuzamosan haladhat egyetlen TCP-kapcsolaton belül, anélkül, hogy várakozniuk kellene egymásra. Ez csökkenti a késleltetést és növeli az átviteli sebességet.
Keretes struktúra: A HTTP 2 kéréseket és válaszokat kisebb bináris keretekre osztja, amelyek függetlenül érkezhetnek és újra összeállhatnak az ügyfél vagy szerver oldalán, így még gyorsabb adatfeldolgozást biztosít.

Sorolja fel a fő HTTP metódusokat és azok célját.

GET: Weboldal olvasása. **HEAD:** Weboldal fejléceinek olvasása.
POST: Weboldalhoz történő hozzáfűzés. **PUT:** Weboldal tárolása. **DELETE:** Weboldal eltávolítása.
TRACE: Bejövő kérés visszaküldése. **CONNECT:** Kapcsolódás proxy-n keresztül.
OPTIONS: Egy oldal opcióinak lekérdezése.

Soroljon fel néhány ismert állapotkódot.

1xx (Információ): A kiszolgáló jóváhagyja az ügyfél kérését, pl. **100** = kérés jóváhagyása.
2xx (Siker): A kérés sikeresen végrehajtódott, pl. **200** = sikeres kérés, **204** = nincs tartalom.
3xx(Átirányítás)oldal átirányítást igényel **301**=oldal áthelyezve,**304**=gyorsítótárban tárolt oldal érvény
4xx (Ügyfél hibája): Az ügyfél kérésében van hiba, pl. **403** = tiltott oldal, **404** = oldal nem található.
5xx (Kiszolgáló hibája): A kiszolgálónál hiba történt, pl. **500** = belső szerverhiba, **503** = próbálkozzon újra később.

Hogyan működik a böngészőbe épített gyorstár?

Letöltött oldalakat a gyorsító tárban tároljuk Ha újra hivatkozunk rá, akkor gyorstárból töltjük be

Hogyan lehet oldal érvényességét ellenőrizni a gyorstárban?

Lejárati fejléc: Ha van, akkor az érvényességi időt mutatja, és a tárolt oldal addig érvényes.
Feltételes GET: A kliens egy kérés során ellenőrzi, hogy az oldal változott-e.
----Ha nem változott, akkor rövid választ kap, és az oldal nem terheli a hálózatot.
----Ha változott, a teljes oldal újra letöltődik, és a gyorstárba kerül.

Ismertesse a proxy szerverek működését, fő feladatait. Mik ezen megoldás korlátai?

Feladatok: Cache: A proxy szerver tárolja a szervezet felhasználói által letöltött oldalakat a gyorstárban. Ha újra hivatkoznak rá, a tárolt oldal gyorsabban töltődik be.
Biztonság: Segít a tűzfalak kezelésében és védelmet nyújt.
Tartalomszűrés: Lehetővé teszi bizonyos weboldalak blokkolását vagy hozzáférésük szabályozását.
Korlátok:Népszerűtlen oldalak: Azokat az oldalakat, amelyek nem népszerűek, nem tárolja a proxy, így ezek nem érhetők el gyorsabban.
HTTPS: A proxy nem férhet hozzá a titkosított HTTPS forgalomhoz, hacsak nem alkalmaznak "man-in-the-middle" (MITM) technikát, ami biztonsági problémákat okozhat.

Hasonlítsa össze a HTTP1.1, HTTP2 és a HTTP3 főbb tulajdonságait. Ismertesse a hatékonyság növelése érdekében bevezetett új megoldásokat.

Protokoll	Bevezetés éve	Hálózati réteg	Fő jellemzők
HTTP/1.1	1997	TCP felett	- TCP felett működik
HTTP/2	2015	TCP felett	- Kérések csővezetéken - Tömörített fejrész - Szerver push (nem kért információt is küld)
HTTP/3	2020 (Internet Draft)	QUIC felett	- QUIC felett működik - A legtöbb böngésző már támogatja

Miért előnyösebb a HTTPS protokoll használata a HTTP-vel szemben?

Problémák a HTTP-vel: Lehallgatás // Üzenetek módosítása // Megszemélyesíté

Ismertesse a HTTPS vázlatos működését. Mire szolgál a kétkulcsos titkosítás? Mire szolgálnak a tanúsítványok?

Kétkulcsos titkosítás (üzenetváltás biztonságos): Nyilvános kulcsot a szerver elküldi
A nyilvános kulcs valódiságát a böngésző ellenőrzi -> Tanúsítvány

Miért van szükség speciális tartalomszállító hálózatokra?

Gyorsabb hozzáférés: A népszerű tartalmakat a felhasználókhoz közel tárolják, így csökkentve a késleltetést.
Több helyszín: Azonos tartalom több példányban elérhető, legközelebbi szerver szolgálja ki
Hatékony terheléselosztás: A kérések optimális elosztása a CDN csomópontok között, elkerülve a túlterheltséget.
DNS alapú irányítás: A CDN a tartalomszolgáltató speciális DNS-szerverei alapján irányítja a felhasználókat a megfelelő szerverhez.

Ismertesse a szerverfarmok működését

Felépítés: Egyetlen logikai webhelyet több kiszolgáló gép szolgál ki.
Kérések kezelése: Az előtétberendezés (**front-end**) fogadja a kéréseket.
A kéréseket elosztja a szerverfarm gépei között (**terheléskiegyenlítés**).
Fontos szempontok: Egy webkérés összes csomagját ugyanarra a kiszolgálóra kell irányítani.
Szűrési és irányítási módszerek: például IP-cím alapján vagy front-end optimalizációval.

1. Foglalja össze röviden a TCP által nyújtott szolgáltatásokat.
    a.) Bájtfolyam
    b.) Megbízható

2. Foglalja össze röviden az UDP által nyújtott szolgáltatásokat.
    a.) Üzenet
    b.) Nem megbízható
3. Hasonlítsa össze az UDP és TCP tulajdonságait.

TCP (bájtfolyam)	UDP (datagramm)
Összeköttetés alapú	Összeköttetés nélküli (datagramm)
A bájtok megbízható módon, sorrendben, 1x kerülnek átvitelre	Üzenet elveszhet, duplikálódhat, sorrendjük keveredhet
Tetszőleges hosszúságú lehet	Korlátos hosszúságú üzenet
Forgalomszabályozás hangolja az adót a vevőhöz	Adó adhat a vevő állapotától függetlenül
Torlódáskezelés hangolja az adót a hálózathoz	Adó adhat a hálózat állapotától függetlenül

Nagyon sok új szolgáltatás

Kb. annyit tud, mint egy IP csomag

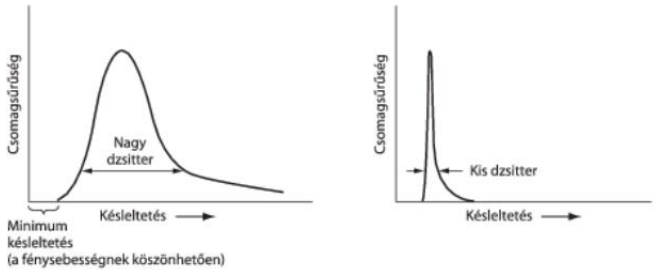
4. Magyarázza el, hogy mire szolgálnak az UDP fejrészben a forrásport és célport mezők.
    a.) Portok azonosítóit tárolják

5. Mit jelent az IP fejrész és mire szolgál? Mi a kapcsolata az UDP ellenőrző összeggel?
    a.) IP fejrész
        i. az IP protokoll által használt adatstruktúra, amely metaadatokat tartalmaz az adatcsomagok továbbításához
        ii. Végpontok azonosítására szolgál (Forráscím és célcím)
    b.) Kapcsolat
        i. Az UDP ellenőrző összeg kiszámításához szükséges az IP fejrész
        ii. Így biztosítva van, hogy az adatok nemcsak az UDP szinten, hanem az IP címzés szempontjából is helyesek maradjanak
6. Mi a socket rendszerhívás célja? Mire szolgál a bind rendszerhívás? Mire szolgál a sendto és recvfrom rendszerhívás? Mit jelent, hogy a recvfrom blokkoló hívás?
    a.) Socket rendszerhívás
        i. célja, hogy létrehozza az alkalmazás és a hálózat közötti kommunikációs végpontot
        ii. socket (hálózati foglalat) objektumot hoz létre, amely lehetővé teszi az adatkommunikációt a hálózaton keresztül
        iii. Támogatja a TCP és UDP protokollt is:
            - TCP: Megbízható kapcsolat alapú kommunikáció
            - UDP: Kapcsolat nélküli, gyors üzenetküldés

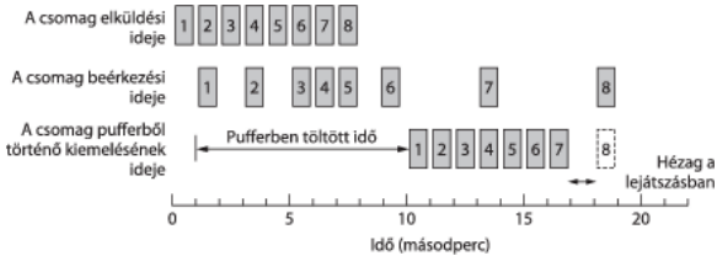
b.) Bind rendszerhívás
    i. egy socket-hez köt egy helyi hálózati címet (IP-cím és port kombinációt)
    ii. biztosítja, hogy a foglalat tudja, melyik hálózati interfészen és porton kell várnia az érkező adatokat
    iii. biztosítja, hogy egy adott IP-címhez és porthoz rendelődve az alkalmazás képes legyen a beérkező kérések fogadására
c.) Sendto
    i. Az üzenetküldéshez használatos, kapcsolat nélküli (UDP) kommunikációban
    ii. Lehetővé teszi, hogy az üzenetet egy adott célcímre és porthoz továbbítsuk anélkül, hogy előzetesen kapcsolatot kellene létrehozni
d.) Recvfrom
    i. adatfogadásra szolgál, kapcsolat nélküli (UDP) kommunikációban
    ii. Lehetővé teszi az érkező üzenetek fogadását, miközben a küldő címét és portját is visszaadja
e.) recvfrom blokkoló hívás
    i. amikor a recvfrom rendszerhívást meghívják, a program végrehajtása megáll (blokkolódik), amíg nem érkezik adat a foglalatra
        - Az adott szál addig vár, amíg a foglalat valamilyen adatot nem kap
        - Ez az alapértelmezett, de átkonfigurálható nem blokkoló módra

7. Mi a távoli eljáráshívás? Mit a jelent a csonk (stub)? Magyarázza el a cliens-oldali és a szerver-oldali csonkok szerepét és működését.
    a.) Távoli eljárás hívás
        i. Célja, hogy úgy viselkedjen, mintha hagyományos (helyi) függvényhívás lenne
    b.) Csonk (stub)
        i. az RPC rendszer fontos komponense, amely a távoli eljáráshívás végrehajtásához szükséges köztes feladatokat végzi el
    c.) Kliens csonk (stub)
        i. Úgy viselkedik, mint egy helyi függvényhívás
        ii. Elvégzi a hálózati feladatokat
            - Paramétereket üzenetbe pakolja (marshaling)
            - Elküldi az üzenetet
            - Veszi a választ
            - Visszatér az eredménnyel a hívóhoz
    d.) Szerver csonk
        i. A távoli függvény tényleges végrehajtását készíti elő, majd az eredményt visszajuttatja a klienshez
        ii. Veszi az üzenetet
        iii. A paramétereket kicsomagolja (demarshaling)
        iv. Meghívja a függvényt
        v. Az eredményeket visszaküldi
8. Mire használjuk az RTP protokollt?
    a.) Szerver több forrást is használhat (pl. video, audio különféle nyelveken)
    b.) Adatokat RTP blokkokba kódolják
    c.) RTP blokkokat UDP-n keresztül továbbítják

9. Mire használjuk az RTCP protokollt?
    a.) Visszacsatolást kezeli:
        i. Hálózat tulajdonságairól a szervernek (késleltetés, jitter, sávszélesség, torlódás)
        ii. Ez alapján képes folyamatosan állítani a minőséget (pl. kódolás, felbontás)
    b.) Szinkronizációt biztosít
        i. A különféle folyamok óráit egymáshoz szinkronizálja
    c.) Forrás-információkat szállít
        i. Pl. a beszélő neve
10. Milyen hatása van a késleltetésnek és a jitternek a valós idejű protokollok működése során?
    a.) Késleltetés
        i. Sok alkalmazásban nincs lényegi hatása (pl. zenehallgatás)
        ii. Egyes alkalmazásokban kis értéken kell tartani (pl. telefon)
    b.) Késleltetés ingadozása (jitter)
        i. Nagyon zavaró! Kompenzálni kell a vevőben!



11. Hogyan kompenzáljuk a jitter hatását a vevőben? Mit jelent a lejátszási pont?
    a.) Pufferelés a vevőben



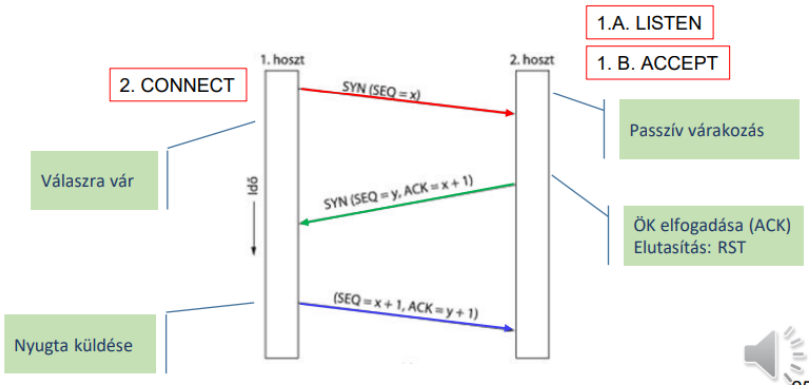
12. Hogyan azonosítunk egy csatlakozót (címzés)?
    a.) Küldő és fogadó létrehoz egy csatlakozót (socket)
    b.) A socket címe három elemből áll:
        i. IP cím (pl. IPv4 vagy IPv6) ←NSAP
        ii. Port száma (16 bites egész) ←TSAP
        iii. Protokoll (pl. TCP, UDP)
13. Mit jelent a „jól ismert port”? + Magyarázza el az inted szerepét és működését.
    a.) A szerverek „Ismert” portokat használnak („Well Known Ports”)
        i. 1024 alatti portok
        ii. Csak privilegizált felhasználóknak
        iii. Minden porthoz egy démon tartozik (pl. ftp démon vagy http démon)
        iv. Gyakran egyetlen felügyelő démont használunk (pl. inetd – Internet Daemon – a Linux alatt)
            - inetd figyel több portot is
            - szükség esetén elindítja a megfelelő démont

- nem kell egyszerre sok démonnak feleslegesen futnia

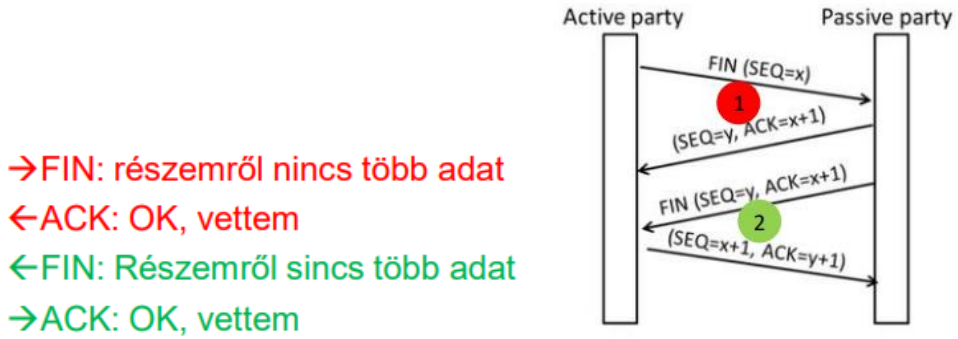
14. Milyen portszámokat kapnak a kliens folyamatok? Kitől kapják a portszámokat?
    a.) Általában ideiglenes portokra csatlakoznak
    b.) Az operációs rendszer választja
    c.) Véletlenszerű
15. Mit jelent, hogy a TCP full-duplex és kétpontos kommunikációt biztosít?
    a.) full-duplex
        i. a TCP-kapcsolatban mindkét fél egyszerre küldhet és fogadhat adatot
        ii. Ez olyan, mintha kétirányú kommunikáció lenne egy időben (például telefonhívásnál mindkét fél beszélhet és hallgathat egyszerre)
    b.) kétpontos kommunikáció
        i. A TCP egy kétpontos (point-to-point) kapcsolatot hoz létre
        ii. egy adott kapcsolat két végpont között valósul meg, amelyeket az IP-címek és portszámok azonosítanak
    c.) Nincs adatszórás (broadcast) vagy többesküldés (multicast)!
16. Mit jelent, hogy a TCP egy bájtfolyamat biztosít?
    a.) nem csomagokra vagy üzenetekre koncentrál, hanem egy folytonos adatfolyamként tekint a küldött adatokra
    b.) küldő alkalmazás bármennyi adatot küldhet egyszerre (akár egyetlen bájtot, akár nagyobb mennyiséget), és a TCP ezt a folyamatot kezeli úgy, hogy a fogadó fél pontosan ugyanazt a sorrendet kapja vissza
    c.) Az adatokat a TCP kisebb darabokra (csomagokra) bontja az átvitelhez, majd a fogadó oldalon újra összerakja őket egy folytonos bájtfolyammá
    d.) Nincs üzenet, nincs üzenethatár
    e.) A szegmensek kontroll infót is szállítanak (pl. ACK) →piggyback
17. Mikor küldi el a TCP a rábízott adatokat? Hogyan lehet gyors továbbítást kikényszeríteni?
    a.) Az elküldendő adatot szabad belátása szerinti időben küldi
        i. Lehet pufferelni, hogy nagyobb szegmens összegyűljön
        ii. PUSH bit: kérheti a TCP-t, hogy ne késleltessen (pl. online játék)
18. A TCP milyen adategységet sorszámoz?
    a.) Minden bájt rendelkezik egy 32 bites sorszámmal
    b.) Ezeket a kapcsolat kezdetén meghatározott kezdeti sorszámhoz viszonyítva kezeli
    c.) Biztosítja a bájtfolyam sorrendiségét, a hiányzó csomagok felismerését és az újraküldési mechanizmus működését
19. Mi korlátozza TCP szegmens méretét?
    a.) IP adatmező
        i. MTU (legnagyobb átvihető adategység) - Pl. Ethernet: 1500B
20. Magyarázza el a TCP fejrészben a sorszám és a nyugtaszám szerepét.
    a.) Sorszám
        i. azt a bájt sorszámát jelöli az adatfolyamban, amely a TCP szegmens első bájtjához tartozik
        ii. a sorszám segít a fogadó félnek abban, hogy a bájtokat megfelelő sorrendben összerakja, még akkor is, ha a csomagok nem sorrendben érkeznek
    b.) Nyugtaszám
        i. jelzi, hogy a fogadó fél melyik bájtot várja következőként az adatfolyamban
        ii. Jelzi, hogy a nyugtázott bájtig az összes adat hiánytalanul megérkezett

iii. a nyugtaszám értéke mindig az utoljára átvett bájt sorszámának egy egységgel növelt értéke (tehát a várt bájt indexe)

21. Magyarázza el a TCP fejrészben az ablakméret mező szerepét.
    a.) A vevőben rendelkezésre álló puffer mérete
22. Magyarázza el a TCP fejrészben a CWR és ECE bitek szerepét.
    a.) Vevő ECE bittel jelez: torlódás van a hálózaton
    b.) Adó CWR bittel jelez vissza, hogy a forgalmat (a torlódási ablaka méretét) lecsökkentette
23. Magyarázza el a TCP fejrészben az ACK, PSH, RST, SYN és FIN bitek szerepét.
    a.) ACK: Jelzi a nyugtaszám érvényességét
    b.) PSH (PUSH): késedelem nélküli továbbítás kérése
        i. Ne legyen pufferelés
    c.) RST (RESET): Összeköttetés helyreállítás (valami baj történt)
    d.) SYN: kapcsolat kiépítés
        i. Jelzi a CONNECTION REQUEST és CONNECTION ACCEPTED üzeneteket
    e.) FIN: kapcsolat bontása
        i. Jelzi, hogy a küldőnek nincs több adata
24. Ismertesse, hogyan alkalmazza a TCP az összeköttetés létrehozásához a háromutas kézfogást.

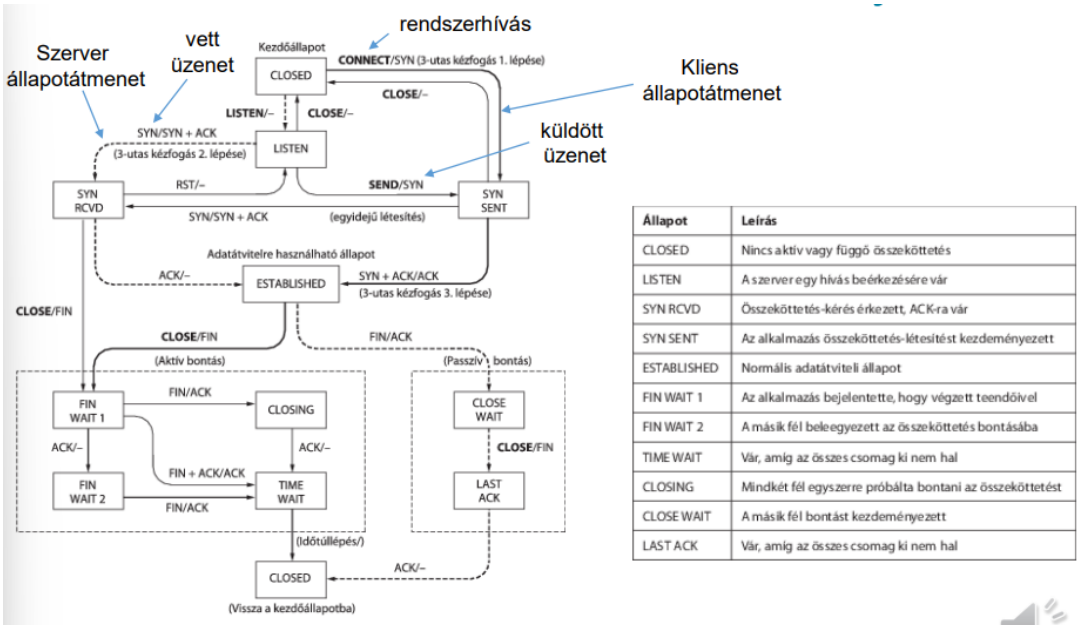


25. Ismertesse a TCP összeköttetés lebontásának módját.
    a.) Mindkét fél bontja a belőle induló kapcsolatot (irányt)
        i. FIN jelzi, hogy nem kíván több adatot küldeni



    b.) Időzítők használata
        i. Ha FIN-re nem jön válasz, bontja az ÖK-t

26. A 27. oldalon található állapotgép segítségével magyarázza el a TCP összeköttetés felépítésének és bontásának menetét.

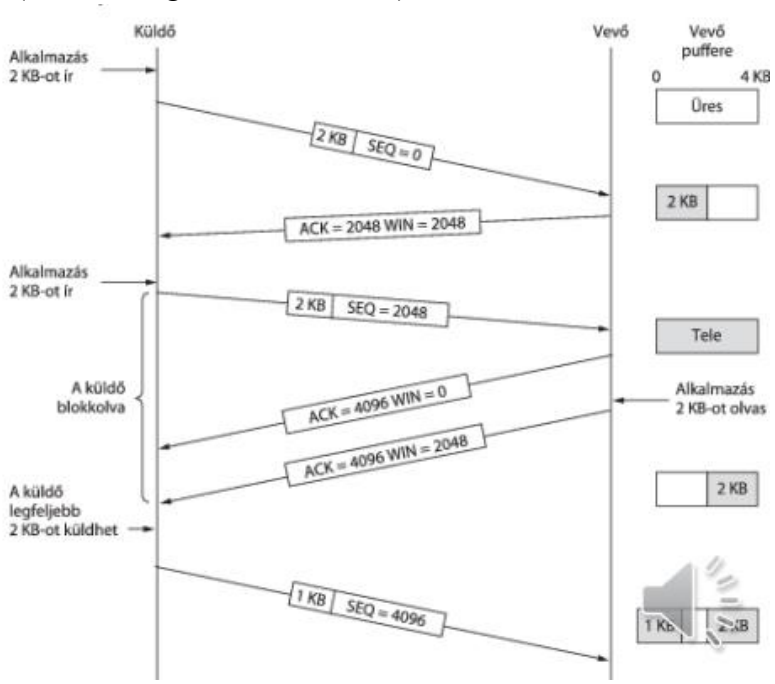


Állapot	Leírás
CLOSED	Nincs aktív vagy függő összeköttetés
LISTEN	A szerver egy hívás beérkezésére vár
SYN RCVD	Összeköttetés-kérés érkezett, ACK-ra vár
SYN SENT	Az alkalmazás összeköttetés-létesítést kezdeményezett
ESTABLISHED	Normális adatátviteli állapot
FIN WAIT 1	Az alkalmazás bejelentette, hogy végzett teendőivel
FIN WAIT 2	A másik fél beleegyezett az összeköttetés bontásába
TIME WAIT	Vár, amíg az összes csomag ki nem hal
CLOSING	Mindkét fél egyszerre próbálta bontani az összeköttetést
CLOSE WAIT	A másik fél bontást kezdeményezett
LAST ACK	Vár, amíg az összes csomag ki nem hal

A TCP összeköttetést kezelő állapotgép
(vastag vonal: normál működés. Szaggatott vonal: szerver, folytonos vonal: kliens)

27

27. Magyarázza el a TCP csúszóablakos forgalomszabályozásának működését.
    a.) Különválik az ACK és a vevő pufferméretének kezelése
    b.) ACK: halmozott nyugta
        i. A nyugtázott bájtig az összes adat hiánytalanul megérkezett
        ii. (ACK a következő – várt – sorszámot tartalmazza)
    c.) WIN: vevőben rendelkezésre álló ablakméret
        i. = maximális küldhető adatmennyiség
    d.) SEQ: sorszám
        i. Az eddig elküldött bájtok száma
        ii. (ez a csomag nem számít bele)



28. Hány bájtot küldhet a kliens a szerver felé, ha a szerver utolsó üzenete a következő volt:
    a.) ACK=67123, WIN=512
    b.) ACK=512, WIN=0

    a.) Mivel a WIN=512, ez azt jelenti, hogy a szerver rendelkezésre álló ablakmérete 512, tehát ennyi adatot küldhet még max a kliens. A kliens küdlött egy ACK-t válaszul tehát még 511 hely van vissza.
29. Egy szervernek 2kB üres pufferterület áll rendelkezésre egy TCP összeköttetés kezelésére. Most a következő (512 adatbájtot tartalmazó) üzenet érkezik a szerverhez:
    ADAT: 512B, SEQ=3000
    a.) Mi lesz a szerver által küldött következő üzenetben az ACK és a WIN értéke? Miért?

    a.) ACK=3512 WIN=1536, mert jelzi, hogy fogadta az 512B adatot, de minden bájthoz egy sorszámot rendel és mivel a 3000-ik sorszámként küldte így a következő, ami jön az a 3000+512=3512-ik sorszám lesz. Jelzi azt is, hogy 2kB (2048B) a puffere mérete, amiből ez az 512B lejön szal 2048-512=1536.
30. Hogyan lehet a szegmens elvesztését időzítő segítségével detektálni? Hogyan célszerű beállítani az időzítő értékét, ha ismerjük az átlagos késleltetési időt (SRTT) és a késleltetési idő szórását (SVAR)?
    a.) Ha a szegmensre nem jön az időzítő lejárta előtt válasz, akkor újraküldjük
    b.) Az időzítő, ha 500ms akkor túl kicsi, ha 900ms akkor túl nagy
    c.) Jobb ötlet: Változtassuk az időzítő értékét az aktuális helyzetnek megfelelően
        i. adaptív időzítő
            - átlag és
            - szórás kell hozzá
31. Hogyan becsüljük a késleltetési idő átlagos értékét? Ismertesse az exponenciális átlagoló működését.
    a.) Exponenciális átlagolással: az aktuálisan mért értékeket és a korábbi simított értékeket kombinálja egy adott súlyozási tényezővel
        i. RTT: mért körülfordulási idő (Round Trip Time)
        ii. SRTT: simított körülfordulási idő (Smoothed Round Trip Time)
        iii. sVAR: az RTT simított „varianciája”

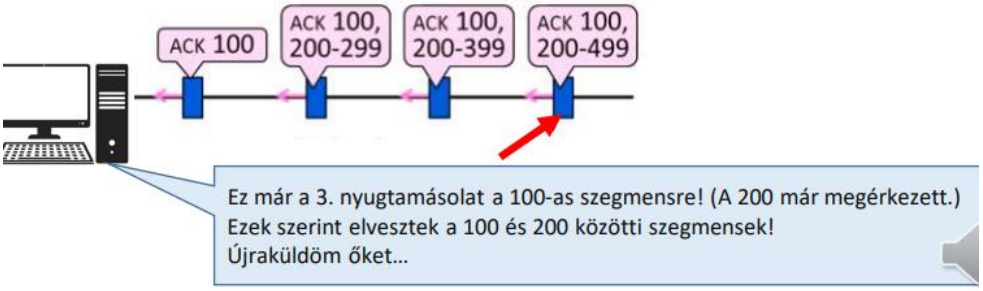
$$SRTT_{\text{új}} = \alpha \cdot SRTT_{\text{régi}} + (1-\alpha)RTT$$

$$sVAR_{\text{új}} = \beta \cdot sVAR_{\text{régi}} + (1-\beta)\left|SRTT_{\text{új}} - RTT\right|$$

$\alpha = 7/8$
$\beta = 3/4$

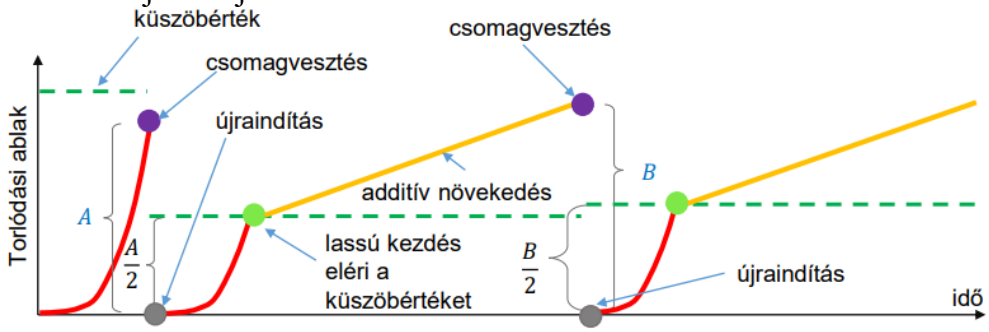
    b.) Alfa és béta súlyozások, a kisebb értékek kisebb, a nagyobb értékek súllyal vannak figyelembevéve

32. Ismertesse a „3 ismételt nyugta” szabályt. Hogyan lehet ennek segítségével a szegmens elveszését detektálni?
    a.) A szegmens valószínűleg elveszett, ha
        i. nem jött rá nyugta RTO időn belül
            - Ehhez ki kell várni az RTO időt
            - Lehetne gyorsabban is detektálni az elveszett csomagot? Igen… 3 nyugtamásolat szabály
        ii. legalább 3 ismételt nyugta (nyugtamásolat) jön egymás után



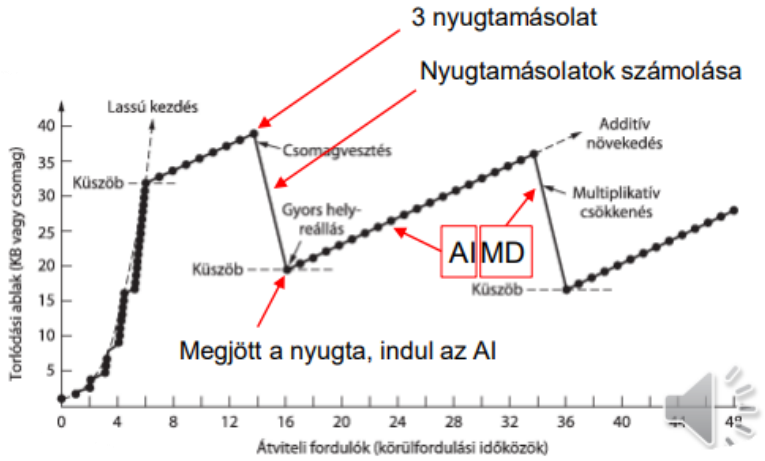
33. Mitől jöhet létre torlódás? Mit jelent a torlódási ablak? Hogyan használja a TCP a forgalomszabályozási és a torlódási ablakot?
    a.) Ha a hálózati terhelés túl nagy
        i. Csomagok feltorlódnak az útválasztókban
        ii. Csomagok késnek, elvesznek
    b.) TCP: torlódási ablakot használ
        i. Mennyi adat lehet a hálózaton egyszerre?
        ii. Hasonló a forgalomszabályozási ablakhoz
            - A TCP együtt használja a két ablakot
            - Amelyik kisebb, annak megfelelő mennyiségű adatot küld ki
34. Ismertesse a nyugtaórajel működését.
    a.) Egy hálózat sebességét a leglassabb szakasza határozza meg
    b.) Egy gyors löketre válaszul visszaérkező nyugták sebessége azt mutatja meg, hogy milyen gyorsan lehet a csomagokat a hálózat leglassabb részén átvinni
    c.) Az adó ennek megfelelő sebességgel ad
        i. Elkerüli a felesleges sorbanállást az útvonalválasztókon
    d.) „sima” kimenőforgalmat biztosít
35. Ismertesse a Lassú kezdés algoritmus működését. Mi célt szolgál ez az algoritmus a TCP-ben?
    a.) Hogy állítsuk be a torlódási ablakot?
        i. Ha nincs torlódás, növeljük (AI)
        ii. Ha torlódás van, csökkentsük (MD)
    b.) A kezdeti munkapont elérése AI-val lassú
        i. Ezért indításkor exponenciálisan növekvő ablakméretet használunk
        ii. Ez lesz a „lassú kezdés” algoritmus (Slow Start)
            - Minden nyugta vételénél
                a. ablakméret eggyel nő
                b. új csomagot küld a régi helyett
                c. új csomagot küld az új üres helyre
            - Úton lévő csomagok száma
                a. RTT alatt duplázódik

36. Ismertesse a TCP Tahoe torlódáskezelését a lassú kezdés segítségével. Használja a 36. oldal ábráját.
    a.) A TCP Tahoe lassú kezdéssel indítja az adatátvitelt
    b.) A lassú kezdésnek van egy küszöbértéke, ennél magasabbra a lassú kezdés nem megy
        i. A küszöbérték kezdetben magas, pl. a forgalomszabályozási ablak mérete lehet
    c.) Ha a lassú kezdésnél torlódást tapasztalunk (csomagvesztés), akkor
        i. visszavesszük a küszöbértéket a jelenlegi torlódási ablak felére és
        ii. újraindítjuk a lassú kezdést
    d.) Ha a lassú kezdés eléri a küszöbértéket, akkor
        i. átkapcsolunk additív növekedésre (AI)
    e.) Ha az additív növekedésnél torlódást tapasztalunk (csomagvesztés), akkor
        i. visszavesszük a küszöbértéket a jelenlegi torlódási ablak felére és
        ii. újraindítjuk a lassú kezdést



37. Mit értünk „gyors újraküldésen” a TCP Tahoe protokollban?
    a.) Torlódás/csomagvesztés érzékelése
        i. Időtúllépés vagy
        ii. 3 nyugtamásolat
            - ekkor a hiányzó csomagokat azonnal újraküldjük
            - Ez a gyors újraküldés (fast retransmission)
38. Magyarázza el a „gyors helyreállítás” működését a TCP Reno protokollban. Használja a 39. oldal ábráját. Hogyan valósítjuk meg TCP-ben az „additív növelés – multiplikatív csökkentés” (AIMD) szabályt?
    a.) A TCP Tahoe továbbfejlesztése a TCP Reno
        i. Ha nyugtamásolatokkal csomagvesztést érzékelünk, az ablakméret felére lépünk vissza (nem pedig 1-re). Ez lesz az MD az AIMD-ben!
        ii. Ezt oldja meg a gyors helyreállítással (fast recovery)
            - 3 csomagmásolat érkezik. Jelenlegi torlódási ablak mérete: T
            - Számoljuk a beérkező nyugtamásolatokat, de küldés felfüggesztve (minden nyugtamásolat azt jelzi, hogy egy csomag időközben sikeresen megérkezett)
            - Addig várunk, amíg T/2 nyugtamásolat érkezik (ekkor éppen T/2 üzenet marad a csővezetékben)
            - ezután minden nyugtamásolatra 1 új üzenetet küldünk
            - Ha megjön a nyugta:
                a. gyors helyreállítás vége
                b. (nyugtamásolatok nem jönnek)

- Additív növeléssel folytatjuk
    a. T/2 lesz az új ablakméret
    b. Küldés indul AI szabállyal



39. Magyarázza el a szelektív nyugtázás működését.
    a.) A szelektív nyugtázás egy **TCP-továbbfejlesztés**, amely hatékonyabbá teszi az adatvesztés kezelését és az adatátvitelt. A szelektív nyugtázás kulcspontjai
        i. Kummulatív (halmozott) ACK mindig van
            - a fogadó (receiver) kummulatív ACK-t küld
                a. azt jelzi, hogy a vevő melyik adatcsomagig (byte-ig) kapta meg a teljes adatfolyamot megszakítás nélkül
        ii. Szelektív ACK (SACK) opcionális (széles körben támogatott)
            - SACK kiegészíti a halmozott nyugtázást, és lehetővé teszi a vevő számára, hogy pontosan jelezze, mely adatcsomagokat kapta meg, még akkor is, ha a sorozatban hiányzó csomagok vannak
40. Magyarázza el az explicit torlódásjelzés működését az ECE és CWR bitek segítségével.
    a.) TCP torlódáskezelés egy további megoldása
    b.) IP útválasztó a csomagban beállítja a torlódás bitet (lásd Differenciált szolgáltatások)
    c.) A vevő a válasz TCP fejrészben beállítja az ECE bitet (ECN Echo)
    d.) A küldő az ECE hatására úgy reagál, mintha csomagvesztés lenne
    e.) A küldő CWR bitet beállítja a következő csomagban (Congestion Window Reduced)