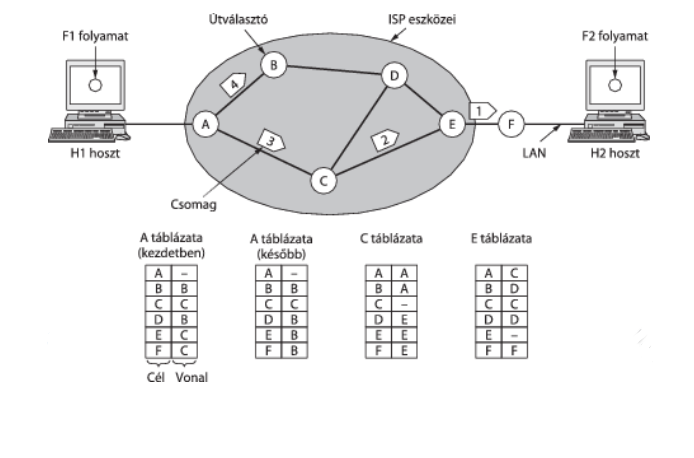
## Mi a hálózati réteg feladata?

* Csomag eljuttatása a forrástól a célig
  + Az adatkapcsolati réteg szolgáltatásait használja
* Esetleg több útválasztón és több hálózaton keresztül
* Ismernie kell a hálózat topológiáját
* Útvonal kiválasztása
  + Optimalizálás
  + Túlterhelés elkerülése

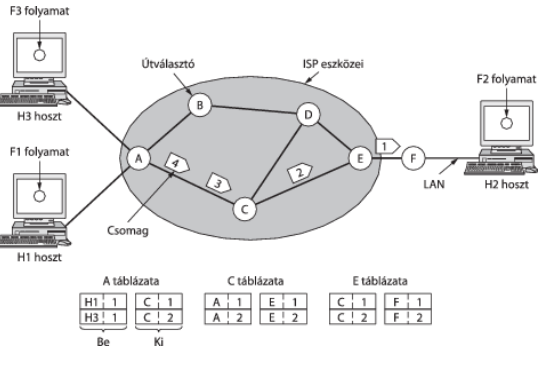
## Ismertesse a Tárol-és-továbbít típusú csomagkapcsolás működését.

* Hoszt:
  + csomagot a szolgáltató felé pont-pont kapcsolaton keresztül küldik a legközelebbi útválasztóig
* Útválasztó tárolja a csomagot, míg be nem érkezik
  + Ellenőrzés (CRC)
* Útválasztó a következő útválasztónak küldi a csomagot
  + Amíg meg nem érkezik a címzett hoszthoz

## Ismertesse a datagram alapú (összeköttetés nélküli) hálózatok működését. Mit tartalmaz az útválasztó táblázat?

* Datagramalapú hálózat
* Nincs előre kiépített útvonal
* Minden csomag egyedi úton halad
* Analógia: levél/távirat
* Példa: IP
* Forrás hoszt:
  + Csomag címzettnek
  + Cím: pl. IP cím
* Útválasztók:
  + Fogad/tárol/továbbít
  + Útválasztó táblázat alapján dönt
  + Ezt az útválasztó algoritmus tartja karban

## Ismertesse az összeköttetés alapú hálózatok működését. Milyen adatokat tartalmaznak az útválasztó táblázatok?

* Virtuálisáramkör-alapú hálózat
* Először útvonal kiépítése, végül lebontása
  + „virtuális áramkör”
  + Ennek azonosítóját tárolják az útválasztók (összeköttetés [ÖK] azonosító)
* Minden csomag egy úton halad
* Végül az útvonal lebontása
* Analógia: telefon
* Példa: MPLS

## Hasonlítsa össze az összeköttetés-alapú és az összeköttetés nélküli hálózatokat az alábbi szempontok szerint: áramkör felépítése, címzés, útválasztás, hibatűrés.

A képen asztal látható

Automatikusan generált leírás

## Ismertesse az útválasztás és a csomagtovábbítás lényegét, a két fogalom közti különbséget. Ismertesse a legrövidebb útvonalat kiválasztó Dijkstra-algoritmus működését.

* Útválasztás
  + Hálózati réteg arról dönt, hogy egy beérkező csomagok merre menjenek tovább
  + Ehhez meg kell tanulni a hálózat aktuális topológiáját
  + A hálózati eszközök együtt, elosztott módon végzik
  + Eredmény:
    - útválasztó táblázatok feltöltése és karbantartása
  + Lassú folyamat
* Csomagtovábbítás
  + Amikor a csomag beérkezik, továbbítás a megfelelő irányba
  + A korábban megtanultak (táblázatok) alapján
  + Gyors
* Dijkstra algoritmusa:
  + keressük egy adott csomóponttól (forrás) a legrövidebb távolságokat (a forrás legyen A)

## Az alábbi hálózatban határozza meg az A pont és az összes többi pont közötti legrövidebb utat a Dijkstra-algoritmus segítségével.

A->D:

3+2+4+2+3=14

3+2+1+6+3=15

7+6+3=16

3+4+4=11

A képen óra, karóra látható

Automatikusan generált leírás

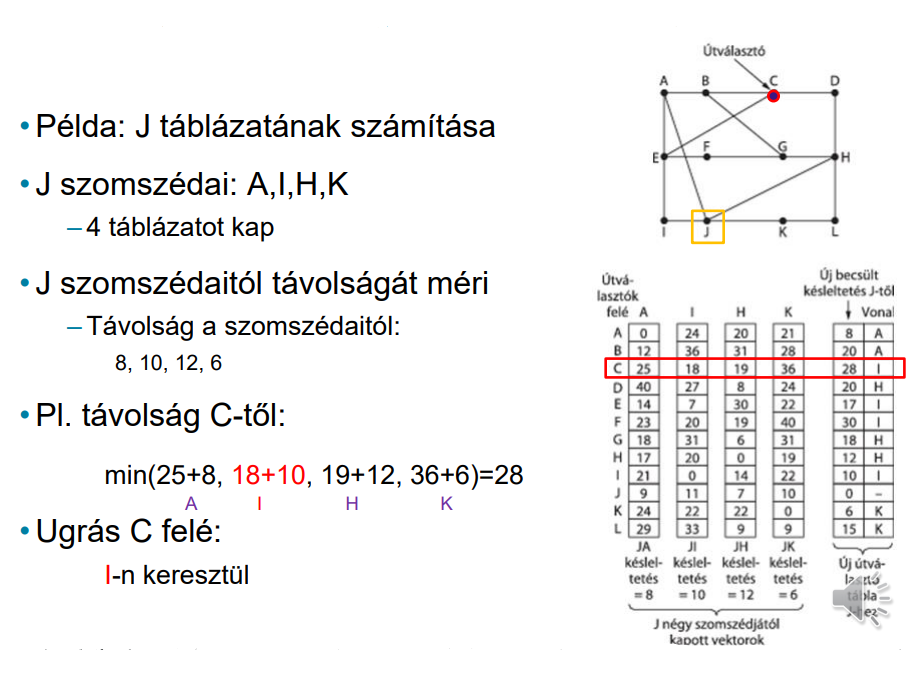
## Ismertesse az elárasztás működését.

* Szabályok:
  + Minden beérkező üzenetet továbbítunk minden szomszédnak
    - Kivéve annak, akitől kaptuk
  + De csak egyszer!
    - Azonosítani kell az üzeneteket (pl. sorszám+feladó azonosítója)
    - Meg kell jegyezni a továbbküldött üzenetek azonosítóját
* Nagyon egyszerű, robosztus
* Nem takarékos
  + Feleslegesen sok üzenet
  + Egy állomás többször is megkaphat egy üzenetet
* Nagyon fontos és gyakori építőelem

## Ismertesse a távolságvektor-alapú útválasztás működését.

* Minden útválasztóban táblázat:
  + Minden célig a legrövidebb távolság
  + Minden célhoz a következő ugrás azonosítója
* A táblázatokat egymással (szomszédokkal) tudatják
* Frissíti saját táblázatát az alábbi források alapján:
  + A szomszédoktól kapott információ (táblázat)
  + Szomszédoktól mért „távolság” (ugrás, késleltetés)
  + Számítás:
    - Távolságom X-től = min(Si szomszédom távolsága X-től + távolságom Si-től)

## Ismertesse a 24. dia példájában, hogy a J csomópont hogyan számítja ki a B, K és F csomópontokhoz tartozó táblázat-bejegyzéseket.



## Ismertesse a távolságvektor-alapú útválasztás kapcsán felmerülő végtelenig számolás problémáját.

* A program beleragad egy végtelen loop-ba, ahonnan nem tud továbblépni.

(

* Probléma: információ esetenként lassan terjed
  + Jó hír gyorsan terjed (van egy új csomópont / új link)
  + Rossz hír lassan terjed (egy csomópont/link megszűnt)
    - Végtelenig számolás problémája

)

## Ismertesse a kapcsolatállapot-alapú útválasztás működését.

* A szomszédokkal való kapcsolatokat használja/cseréli (Link State Routing)
* Minden útválasztó tennivalója (5 lépés):
* Felkutatni a szomszédait és megtudni a hálózati címeiket.
* HELLO üzenetek minden linken
* Beállítani a távolság vagy a költség értékét a minden szomszédjáig a mért késleltetés alapján.
* A költség a sávszélesség reciprokával arányos
* Mérés: pl. ECHO körülfordulási ideje
* Összeállítani egy csomagot, amely a most megtudottakat tartalmazza.
* Tartalmaz még sorszámot és kort is.

## Az ábrán látható hálózatban töltse ki az egyes csomópontok kapcsolat-állapot csomagjait.

A képen szöveg, elektronika, számológép, billentyűzet látható

Automatikusan generált leírás

## Az alábbi kapcsolat-állapot csomagok alapján rajzolja fel a hálózatot és határozza meg az A és D csomópontok közötti legrövidebb utat.

A képen szöveg, elektronika, számológép, billentyűzet látható

Automatikusan generált leírás A képen óra látható

Automatikusan generált leírás

## Milyen problémái vannak a kapcsolatállapot- és távolságvektor-alapú algoritmusoknak? Milyen körülmények között alkalmazhatók?

* Teljes hálózat topológiája kell hozzá
* Nagy hálózatokra nem alkalmasak

## Ismertesse a hierarchikus útválasztási algoritmusok működését.

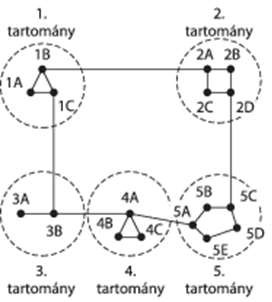
* Alapötlet: a hálózatot régiókra kell osztani
* Útválasztás két lépesben:
  + Régiók között
  + Régión belül
* Analógia:
  + Levéltovábbítás
  + Telefonközpontok

## Az ábrán látható hierarchikus hálózatban határozza meg a 2B csomópont hierarchikus útválasztó tábláját.

2B->3

2B-2A-1B-1C-3

2B|2A|4



A képen asztal látható

Automatikusan generált leírás

## Ismertesse az adatszórás működését. Milyen módszereket alkalmazhatunk?

* (Broadcasting)
* Cél:
  + a hálózat minden csomópontjához eljuttatni az üzenetet
* Legegyszerűbb megoldás:
  + elárasztás
* Egyéb módszerek
  + Feszítőfa alkalmazása

## Ismertesse a többesküldés működését. Milyen megvalósítási módszereket alkalmazhatunk?

* (Multicasting)
* Cél:
  + a hálózat egy csoportjához eljuttatni az üzenetet
  + A csoportnak van egy közös címe
* Lehetséges megoldások
  + Elárasztás + vevőben szűrés
    - Jó, ha sűrű a hálózat
    - Pazarló, hiszen oda is eljut, ahová felesleges
  + Csonkolt feszítőfa alkalmazása
    - Feszítőfa építése a forrásból
    - Felesleges részek levágása

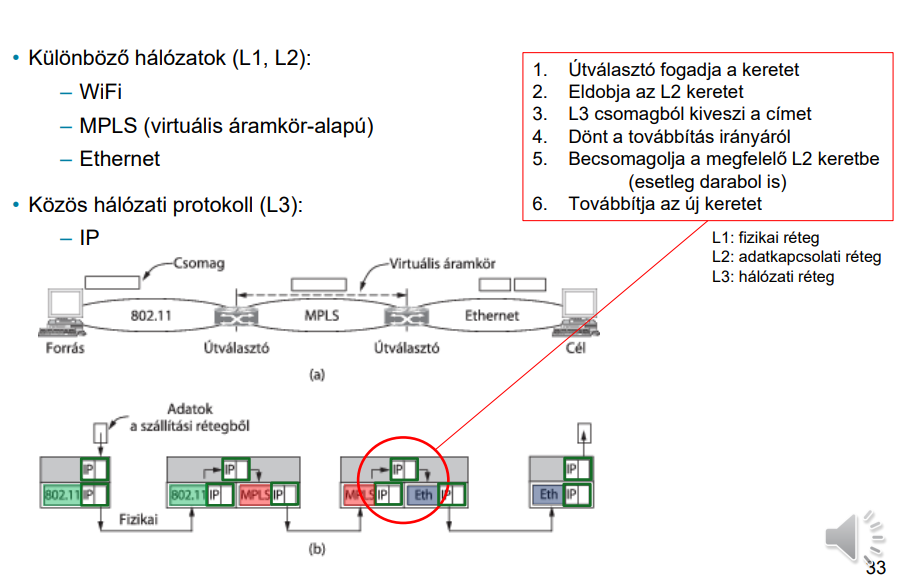
## Ismertesse a bárkinek-küldés működését. Milyen módon valósítható meg?

* (Anycasting)
* Cél:
  + a hálózat egy csoportján belül…
  + … egy tetszőleges (legközelebbi) taghoz kell eljuttatni az üzenetet
  + A csoportnak van egy közös címe
* Megoldás:
  + Minden csoporttagnak ugyanaz a címe
  + A távolság- vagy kapcsolatállapot-alapú útválasztás mindig a legközelebbit választja ki

## Ismertesse az útválasztó működését, amely két különböző hálózatot köt össze (pl. MPLS és Ethernet).

* Útválasztó fogadja a keretet
* Eldobja az L2 keretet
* L3 csomagból kiveszi a címet
* Dönt a továbbítás irányáról
* Becsomagolja a megfelelő L2 keretbe (esetleg darabol is)
* Továbbítja az új keretet

## Rajzolja fel az útválasztóba beérkező és az onnan távozó keretek felépítését.



## Ismertesse az alagút típusú átvitel működését.

* Gyakori probléma:
* Forrás és célhosztok azonos típusú hálózaton vannak
  + Pl. egy cég két telephelye IPv6
* A közbülső hálózat más típusú
  + Pl. IPv4.
* Alagút (tunnel):
  + Becsomagoljuk a forrás csomagot a közbülső hálózati réteg csomagjába
  + Átvitel
  + Kicsomagolás
  + Az eredeti protokoll szerint halad tovább

## Miért szükséges a csomagok tördelése?

* Csomagok mérete különböző
* Hálózatok maximális csomagmérete (MTU) is változik
* Mit tegyünk, ha nagy a csomag, de „keskeny” a hálózat
  + Csomagok tördelése
* Hol tördeljünk?
  + A feladó tördel MTU szerint
  + Az útvonalválasztók tördeljenek igény szerint

## Ismertesse az átlátszó darabolás működését. Mik az előnyei és hátrányai ennek a megoldásnak?

* Minden hálózatban a belépéskor darabolás, kilépéskor összeállítás
* Sok munka az útválasztóknak

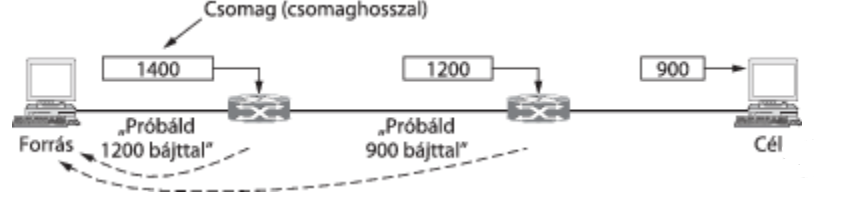
## Ismertesse a nem átlátszó darabolás működését. Mik az előnyei és hátrányai ennek a megoldásnak?

* Minden hálózatban a belépéskor darabolás, de kilépéskor nincs összeállítás
* A csomagot a címzett állítja össze
* Egyszerű (pl. IP)

## Mit jelent az MTU?

* Hálózatok maximális csomagmérete

## Hogyan tudja meghatározni egy forrás csomópont az MTU-t??????



## Mit jelent az belső átjáró protokoll? Mi a célja? Nevezzen meg példákat. Melyik módszert használjuk elterjedten?

* Cél: minél hatékonyabb útvonalkeresés saját hálózaton belül (intradomain routing)
* RIP (Routing Information Protocol)
  + Távolságvektor-alapú
  + Végtelenig számolás problémája
* OSPF (Open Shortest PathFirst)
  + Kapcsolatállapot-alapú
  + A leggyakoribb belső átjáró protokoll
* IS-IS (Intermediate System to Intermediate System)
  + OSPF elődje

## Mit jelent a külső átjáró protokoll? Mi a célja? Nevezzen meg konkrét protokollt.

* Cél: AS-ek közötti útvonalkeresés
* BGP (Border Gateway Protocol)

## Milyen szolgáltatásokat különböztetünk meg a külső átjáró protokollokban? Ismertesse ezek jelentését.

* Tranzit: átmenő forgalom az internet másik oldal felé (pl. ügyfél-ISP, ISP-ISP)
* Peering: az egymás közötti forgalom ingyenes (ISP-ISP)

## Ismertesse a BGP működését. Mit tartalmaz az útvonal-vektor? Hogy épül fel egy útvonal-vektor?

* BGP útvonal-vektort használ
* Cél prefix – Útvonal (AS-ek sorozata) – következő ugrás
* A BGP az útvonal-vektorokat terjeszti a hálózatban az útválasztókon keresztül
* Az AS határán az útválasztó a lista elé teszi az AS számát, a következő ugrásnak pedig magát állítja be
* A határ-útválasztók egymás útvonalait is megtanulják
* Útválasztás a kapott útvonal-vektorokat alapján történik

## Ismertesse a BGP-ben gyakran alkalmazott stratégiákat.

* Peering útvonalaknak előnye van a tranzittal szemben
* Rövidebb útvonal jobb
* AS-en belül a legolcsóbb (legrövidebb) útvonal használata
  + Ez a korai kilépés, vagy forró krumpli útválasztás
  + Pl. az A-C csomagok a felső útvonalon fognak haladni, a B-C csomagok pedig az alsón
* A határon csak olyan útvonalat hirdet, amelyet a szomszéd AS felé támogat
  + Többit kiszűri (akkor is, ha ilyen útvonal létezik)

## Ismertesse az IPv4 fejrész egyes mezőinek feladatát (ábra alapján).

* Verzió: 4
* IHL:
  + fejrész hossza 32 bites szavakban
* Differenciált szolgáltatások:
  + 6 bit: szolgáltatási osztályok (pl. gyorsított, biztosított)
  + 2 bit: explicit torlódásértesítés
* Teljes hossz:
  + Fejrész + adatrész, max. 65535 bájt
* Azonosítás:
  + Darabolás esetén a datagramot azonosítja
  + Azonos datagramhoz tartozó darabok azonosítója ugyanaz
* Nem használt bit
* DF (Don’t fragment):
  + Darabolás tiltása
  + Hasznos az MTU (legnagyobb átvihető adategység) felderítésére
* MF (More fragments):
  + Minden darabban 1, kivéve az utolsót
* Darabeltolás:
  + Darabolás esetén a jelen darab pozíciója a datagramban
* Élettartam:
  + Ugrásokat számolja, minden ugrásnál értéke csökken eggyel
  + Ha nulla, a csomagot el kel dobni
  + Megelőzi, hogy hiba esetén a csomagok végtelen ideig kószáljanak
* Protokoll:
  + Melyik szállítási protokollnak kell adni a csomagot?
  + Pl. TCP, UDP
* Fejrész ellenőrző összeg:
  + 16 bites ellenőrző összeg
  + Minden ugrásnál számítani kell Λ (hiszen az élettartam változik)
* Forráscím, célcím:
  + 32 bites IPv4 címek
* Opciók:
  + Azonosító, hossz, adat
  + Ritkán használt funkciók számára
    - nem kell állandó helyet a fejrészben fenntartani
  + Ritkán használják

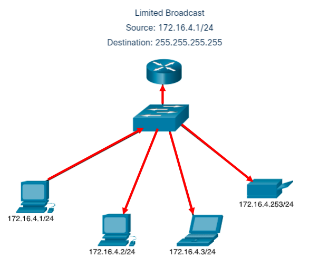
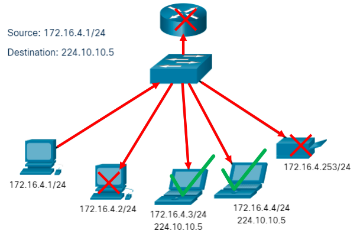
## Ismertesse az IPv4 címek felépítését. Mit jelent a prefix, a hoszt-rész, az alhálózati maszk és a prefix hossz?

* 32 bites
* Jelölés:
  + Pontokkal elválasztott decimális jelölés. Pl. 193.224.41.159
* Hálózati rész
  + előtag (prefix)
    - A hálózat folyamatos címblokkja
* Állomás (hoszt) rész
* Alhálózati maszk:
  + Egyesek a hálózati rész helyén, nullák az állomás részen
  + Pl.: 255.255.255.0
* Prefix hossz jelölés: egyesek száma az alhálózati maszkban
  + Pl. /24, /25, /8, stb.

## A 123.45.67.89/24 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?

* Decimális: 123.45.67.89
* Bináris: 01111011 00101101 01000011 01011001
* Bináris alhálózati maszk: 11111111 11111111 11111111 00000000 (24db 1-es)
* Alhálózati maszk: 255.255.255.0
* Prefix hossz: /24
* Röviden: 123.45.67.89/24
* Hálózat címe: 123.45.67.0
* Első host cím: 123.45.67.1
* Utolsó host cím: 123.45.67.254
* Hálózati broadcast cím: 123.45.67.255
* Hány állomás lehet a hálózaton? 28 − 2 = 254

## Mit jelent az IPv4-en az unicast, broadcast és multicast? Hogyan működnek?

* Unicast (egyesküldés)
  + Egyetlen címzett van
  + Pl: 192.168.1.18
* Broadcast (üzenetszórás)
  + Mindenki címzett az adott hálózaton
  + A lokális hálózaton: 255.255.255.255
  + Célzott broadcast: távoli hálózati cím + csupa 1 bit
    - Pl. 192.168.1.255 (/24)
    - Ezt általában tiltják
* Multicast (többesküldés)
  + Egy forrásból egy csoportba küldhető üzenet
  + Fel kell „iratkozni” a csoportba
  + 224.0.0.0 és 239.255.255.255 közötti tartomány van fenntartva multicast címekre

## Ismertesse a privát IP-címek szerepét és használatát az IPv4-ben.

* Privát IP címek
  + Hálózaton belüli címzésre
    - 10.0.0.0/8 (10.0.0.0 - 10.255.255.255)
    - 172.16.0.0/12 (172.16.0.0 - 172.31.255.255)
    - 192.168.0.0/16 (192.168.0.0 - 192.168.255.255)
  + Hálózatok között ilyenkor NAT kell

## Mit jelent a loopback-cím, mire használjuk?

* Loopback címek
  + 127.0.0.0 /8 (127.0.0.1 - 127.255.255.254)
  + Címzett: a küldő hoszt
  + Tesztelésre használjuk

## Mire szolgálnak az autokonfigurációs címek?

* Link-local (autokonfigurációs) címek
* 169.254.0.0 /16 (169.254.0.1 - 169.254.255.254)
* Automatic Private IP Addressing (APIPA)
* Pl. Windows DHCP kliens használja, ha nincs elérhető DHCP szerver

## Ismertesse az alhálózatok létrehozásának módszerét egy közös címtartományon belül.

* Pl. Egy egyetemnek rendelkezésére áll egy /16 címtér: 128.208.0.0/16
  + Ez 16 bites host címet jelent -> 216 − 2 = 65534 darab host
  + Ezt 3 tanszék részére kell felosztani (3 alhálózat):
    - Számítástudományi Tanszék (SZT) : /17 (címek fele)
    - Villamosmérnöki Tanszék (VT) : /18 (címek negyede)
    - Bölcsészettudományi Tanszék (BT) : /19 (címek nyolcada)
    - Nincs kiosztva (tartalék): címek nyolcada
* Számítástudományi Tanszék: 10000000 11010000 1|xxxxxxx xxxxxxxx 32768-2 db cím
* (Maradék: 10000000 11010000 0|xxxxxxx xxxxxxxx) 32768-2 db cím
* Villamosmérnöki Tanszék: 10000000 11010000 00|xxxxxx xxxxxxxx 16384-2 db cím
* (Maradék: 10000000 11010000 01|xxxxxx xxxxxxxx) 16384-2 db cím
* Bölcsészettudományi Tanszék: 10000000 11010000 011|xxxxx xxxxxxxx 8192-2 db cím
* (Tartalék: 10000000 11010000 010|xxxxx xxxxxxxx) 8192-2 db cím

## 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?

128.129.130.0/25

128.129.130.128/25

## 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?

FO: 128.129.131.0/25

AD: 128.129.131.128/26

KO: 128.129.131.192/26

## A fenti vállalathoz érkezik a 128.129.131.173 címre egy csomag. Hogyan kell eldönteni, hogy melyik alhálózatra továbbítsuk?

A fenti címek alapján eldöntjük, melyik tartományba esik bele. (AD)

## Mit jelent az előtagok csoportosítása? Milyen előnnyel jár?

## Hogyan működik a Leghosszabb egyező előtag útválasztás?

* Elvi működés:
  + Csomag beérkezik, a címe A
  + Útválasztó megnézi, hogy az A cím melyik bejegyzésére illeszkedik
  + Ha több ilyen bejegyzés is van, akkor kiválasztja azt, amelyik a leghosszabb előtaggal bír
    - Pl. ha egy /22 és egy /19 bejegyzésre is illeszkedik, akkor a /22 bejegyzést használja
  + A bejegyzésnek megfelelő irányba küldi tovább a csomagot

## 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?

194.24.0.176: London

194.24.0.78: Gézaháza-alsó

## Mit jelent a CIDR?

Osztály nélküli körzetek közötti útválasztás

## Miért van szükség NAT-ra? Ismertesse a NAT működését.

* Kevés az IPv4 cím
* A szolgáltató csak 1 címet ad az ügyfélnek
* Az ügyfélnek sok címre lenne szüksége
* Megoldás: NAT (Network Address Translation)
  + Hálózaton belül egyedi címek
  + Privát címtartományból:
    - 10.0.0.0 – 10.255.255.255/8 (16 777 216 hoszt)
    - 172.16.0.0 – 172.31.255.255/12 (1 048 576 hoszt)
    - 192.168.0.0 – 192.168.255.255/16 (65 536 hoszt)
* Csak korlátozásokkal működik
  + TCP, UDP

## 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?

## Miért nem univerzális megoldás a NAT?

* IP címek nem globálisan egyediek
  + pl. sok 10.0.0.1 cím a hálózaton
* Nem tud bárki bárkinek küldeni
  + NAT mögül csak kezdeményezni lehet
  + További trükkök kellenek ennek áthidalására
* Az összeköttetés nélküli internetbe összeköttetés-alapú kapcsolatot kever
  + NAT tábla sérülése: összeomlás
* Összemossa a 3. és 4. Réteget
  + portszámok: 4. Réteg
* TCP-n és UDP-n kívül más szállítási protokollok is vannak…

## Mik az IPv4 és IPv6 közötti legfőbb különbségek?

* Sokkal több IP cím van (32 bites helyett 128 bites cím)
* Fejrész egyszerűbb lett (13 mezőből 7 lett)
* Opciók jobb támogatása
  + Szükségszerű a rövidebb fejrész miatt
  + Gyorsabb csomagfeldolgozást tesz lehetővé
* Biztonság javítása
* Szolgáltatásminőség nagyobb hangsúlyt kapott
  + Multimédia

## Ismertesse az IPv6 fejrész egyes mezőinek feladatát (ábra alapján).

* Verzió: 6
* Differenciált szolgáltatások:
  + hasonlóan az IPv4-hez.
  + torlódásjelzés, prioritás
* Folyamcímke:
  + Virtuális-áramkör alapú megközelítést tesz lehetővé
  + Folyamot előre fel lehet állítani, ez az azonosítója
  + Minden útválasztó kikeresi táblázatából, hogy a címke milyen különleges elbánást igényel
* Adatmező hossza:
  + Fejrész utáni méret
* Következő fejrész:
  + Jelzi, hogy van-e következő opcionális fejrész, és milyen típusú
  + Ha ez az utolsó IP fejrész, akkor itt jelzi, hogy melyik szállítási protokollnak kell a csomagot adni (TCP, UDP)
* Ugráskorlát:
  + Mint az IPv4 élettartam

## Mit jelent az IPv6-ban az unicast, multicast és anycast? Hogyan működnek? Mit jelent a GUA és LLA?

* Unicast
  + Egyetlen címzett
  + Globális (Global Unicast Address - GUA):
    - mint IPv4, globálisan egyedi. (Opcionális)
    - 2000::/3 (első 3 bit: 001)
  + Lokális (Link-Local Address - LLA):
    - csak helyi hálózatra alkalmazzuk. (Kötelező)
    - fe80::/10
  + Localhost:
    - ::1/128
* Multicast
  + Több címzett, prefix: ff00::/8
  + Pl.: ff02::1 – minden hoszt, ff02::2 – minden router (azonos adatkapcsolaton)
* Anycast
  + Unicast üzenet (egy cím), de a címzett bármely lehet a lehetséges címzettek közül (általában a legközelebbi)

## Hogyan lehet IPv6 alatt alhálózatokat létrehozni?

## Milyen részekre tagozódik az IPv6 cím?

* 128 bit van
* Hoszt címbitek: 64 bit
* Prefix: összesen 64 bit
  + Ebből a 16 alsó bit az alhálózatokat különbözteti meg
  + A 48 felső bit a globális útvonalválasztási előtag

## Ismertesse az IPv6 címek felépítését, jelölését.

* Forráscím, célcím
  + 128 bit (16 bájt)
  + Formátum:
    - 8 csoport
    - Kettősponttal elválasztva
    - Mindegyik csoportban 4-4- hexadecimális számjegy (hextet)
    - Pl.:8000:0000:0000:0000:0123:4567:89AB:CDEF
  + Egyszerűsítések a jelölésben:
    - Csoporton belül a bevezető 0-k elhagyhatók
      * Pl.: 0123 –> 123
  + Csupa nulla csoportok (egy vagy több) két kettősponttal helyettesíthető
    - csak egyszer egy címben
    - Pl.: 8000::123:4567:89AB:CDEF
  + IPv4 címek írásmódja:
    - ::192.31.20.46

## Írja fel teljes alakban (rövidítés nélkül) a következő IP-címeket. Azonosítsa a címekben az útválasztási előtagot, az alhálózat azonosítóját és a hoszt interfész azonosítóját.

               8000:1::123:67:89AB:CDE,

8000:0001:0000:0000:0123:0067:89AB:0CDE

               2001::1

2001:0000:0000:0000:0000:0000:0000:0001

               2001:db8:12:567:1::12

2001:0DB8:0012:0567:0001:0000:0000:0012

               fe80::

FE80:0000:0000:0000:0000:0000:0000:0000

               ::1

0000:0000:0000:0000:0000:0000:0000:0001

               2001:2:3:4:5:6:7:8

2001:0002:0003:0004:0005:0006:0007:0008

## Sorolja fel az IPv4 és IPv6 vezérlő protokolljait. Ismertesse a szerepüket.

* IPv4
  + ICMP - Internet Control Message Protocol
    - Internetes vezérlőüzenet protokoll
  + ARP - Address Resolution Protocol
    - Címfeloldási protokoll
  + DHCP - Dynamic Host Configuration Protocol
    - Dinamikus hosztkonfigurációs protokoll
* IPv6
  + ICMPv6
    - IPv6 verzió, a fenti 3 funkció egyben

## Ismertesse az ICMP(v4) legfontosabb üzenettípusait. Hogyan működik és milyen üzeneteket használ a ping és tracert?

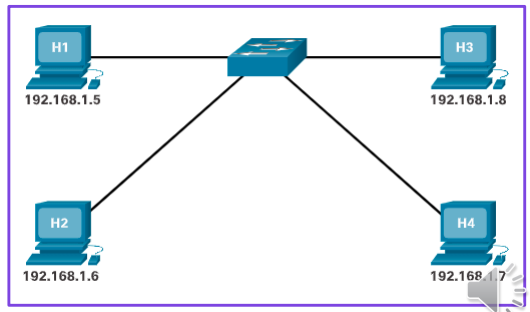
* Internetes vezérlőüzenet protokoll (Internet Control Message Protocol)
* Váratlan események jelzésére, tesztelésre

A képen asztal látható

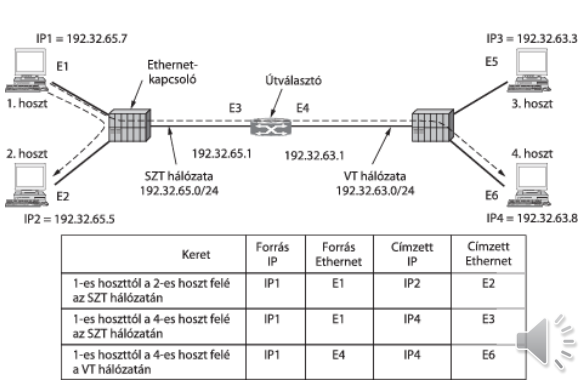
Automatikusan generált leírás

## Ismertesse az ARP üzeneteit és a protokoll működését.

* Címfeloldási protokoll (Address Resolution Protocol)
* Probléma:
  + Hálózati réteg logikai (IP) címeket használ
  + Hálózati réteg az adatkapcsolati réteg szolgáltatásait használja
  + De az adatkapcsolati réteg fizikai (MAC) címeket használ
  + Honnan tudjuk, hogy melyik fizikai címhez melyik logikai cím tartozik?
* Megoldás:
  + A képen szöveg látható

    Automatikusan generált leírásARP
* ARP REQUEST: Broadcast keret küldése az adatkapcsolati rétegben:
  + Helló mindenki (FF-FF-FF-FF-FF-FF). Kié ez az IP-cím?
  + A fizikai címem AAA (ide kérem a választ).
* ARP REPLY: Válasz a feladónak (unicast):
  + Helló AAA. Enyém ez a cím. A fizikai címem BBB.
* Üzenet küldése:
  + Ethernet keret felépítés
    - (forrás: AAA, cél: BBB)
  + Üzenet elküldése
* Üzenet vétele:
  + Ethernet keretet veszi a BBB című állomás
  + Keretet leveszi, a csomagot átadja az hálózati rétegnek.

## Ismertesse az ARP táblázat szerepét és használatát.

* ARP használata:
  + Hálózaton belüli cím: másik hoszt fizikai címe
  + Hálózaton kívüli cím: átjáró fizikai címe
* Optimalizálás:
  + ARP táblázat
  + Ismert címeket tartalmazza:
    - IP – MAC
  + Idővel „lejár”
  + Működés:
    - Küldés előtt ellenőrzi a táblát
    - Ha van bejegyzés, használja
    - Ha nincs, ARP REQUEST
    - ARP REPLY eredményét beírja

## 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?

* Dinamikus hosztkonfigurációs protokoll (Dynamic Host Configuration Protocol)
* Probléma:
  + Mi az IP címem?
* Megoldás:
  + Kézzel konfiguráljuk
  + Automatikusan osztjuk ki: DHCP
* DHCP:
  + IP címeket „lízingel”: meghatározott időre
  + Egyéb paraméterek:
    - Hálózati maszk
    - Alapértelmezett átjáró
    - DNS szerver…
  + Valójában egy alkalmazás UDP felett
* IP cím kérése:
  + DHCP DISCOVER: Broadcast kérés
    - Helló, van itt egy DHCP szerver? Kérek egy IP címet.
  + DHCP OFFER: Unicast válasz
    - Hello, itt a DHCP szerver. Mit szólnál ehhez a címhez?
  + DHCP REQUEST: Broadcast kérés
    - Köszönöm, kérem ezt a címet
  + DHCP ACK: Unicast válasz
    - Rendben van. 12 órára a tiéd.
* IP cím frissítése (lízing lejárta előtt):
  + DHCP REQUEST: Broadcast kérés
    - Szeretném újra ezt a címet
  + DHCP ACK: Unicast válasz
    - Rendben van. 1 órára a tiéd.
* DHCP információk (ipconfig /all)

## Mire szolgál az ICMPv6? Mire szolgál a Neighbor Discovery Protocol (ND)?

* ICMP-hez hasonló funkciók
  + Hiba, visszhang, időtúllépés, …
* Neighbor Discovery protocol (ND)
  + Neighbor Solicitation (szomszéd megszólítása)
  + Neighbor Advertisement (szomszéd hirdetés)
  + Router Solicitation (útválasztó megszólítása)
  + Router Advertisement (útválasztó hirdetés)
  + Redirect Message (üzenet átirányítás)

## Ismertesse a NS és NA üzenetek szerepét, a szomszédságfelderítés menetét.

* IPv6-ban hasonló a működés az ARP-hoz:
  + Szeretnék egy IPv6 címre üzenetet küldeni.
  + Tudom-e a MAC címét?
    - Neighbor Cache: mint ARP tábla
    - Ha van benne infó, akkor használjuk (GOTO 5), ha nincs, akkor GOTO 3
  + NEIGHBOR SOLOCITATION (NS): Multicast keret az adatkapcsolati rétegben:
    - Helló szomszédok, akinek ehhez hasonló IPv6 címe van. Kié ez az IP cím?
    - A fizikai címem AAA (ide kérem a választ).
  + NEIGHBOR ADVERTISEMENT (NA): Válasz a feladónak (unicast)
    - Helló AAA. Enyém ez a cím. A fizikai címem BBB.
  + Üzenet küldése
  + Üzenet vétele
    - Neighbor cache frissítése

## Ismertesse az RS és RA üzenetek használatát. Milyen információt hordoz az RA üzenet? Milyen módokon juthat egy hoszt IPv6 címhez?

IPv6 DHCP-szerű funkciója a globális IP-cím (GUA) előállítására

* ROUTER SOLOCITATION (RS): Multicast üzenet minden útválasztónak
  + Hello útválasztók (ff02::2), cím információra van szükségem
* ROUTER ADVERTISEMENT (RA): Multicast üzenet minden állomásnak
  + Helló minden hoszt (ff02::1), küldöm az információt:
    - Hálózati prefix és a prefix hossza
    - Alapértelmezett átjáró címe
    - DNS címe
    - [DHCP szerver címe]
  + De mi legyen a hoszt cím (a prefix már megvan)?
    - Csinálj magadnak egyet (SLAAC)
    - Küldök egy DHCP szerver címet, konzultálj vele

## Mit jelent a SLAAC? Ismertesse a működését.

SLAAC: Stateless Address Autoconfiguration

## Mi a szállítási réteg feladata? Milyen szolgáltatásokat nyújt a hálózati réteg?

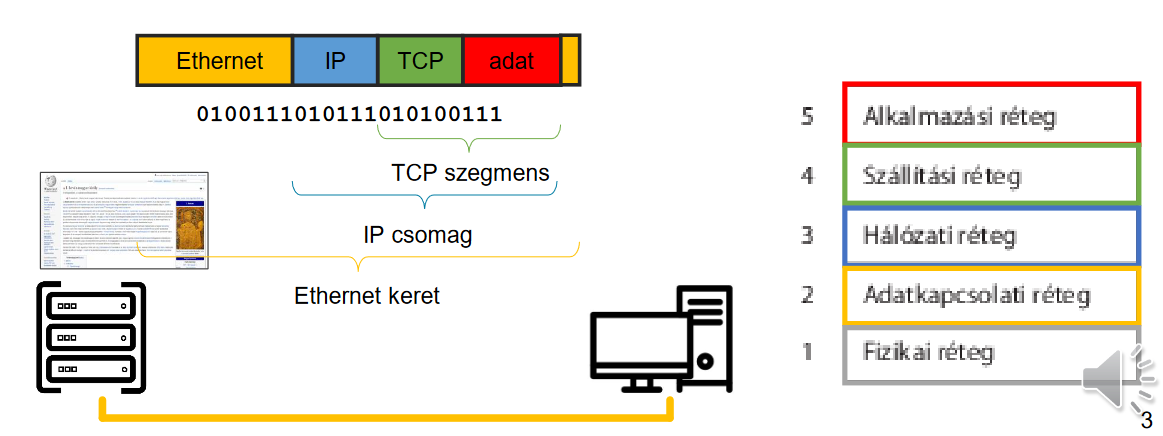
Feladata:

* Forrásgép egy folyamatától a célgép egy folyamatáig
* Megbízható szolgáltatás
* Használja a hálózati réteg szolgáltatásait
* A felhasználó gépén fut
  + Teljes felhasználói kontroll
  + Társentitások tudnak egyeztetni
    - Pl.: megérkezett? Ha nem, újraküldés.
  + Így megbízhatóbb tud lenni, mint az alatta lévő réteg

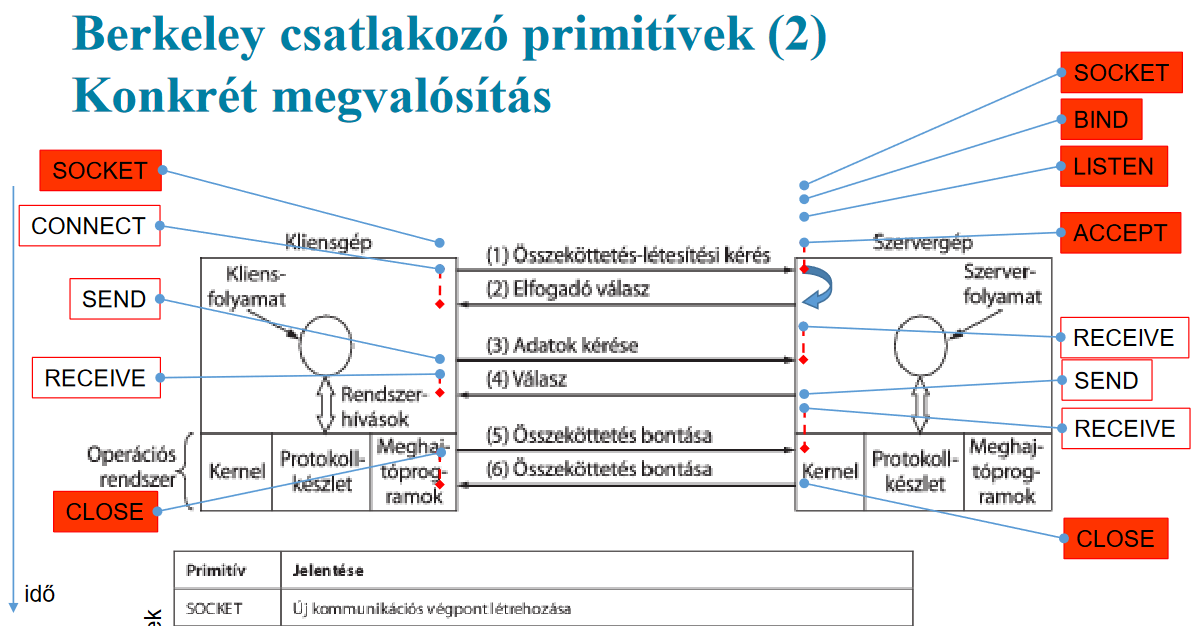
Szolgáltatások:

* A felsőbb rétegeknek nyújtott szolgáltatások:
* Összeköttetés alapú
* Összeköttetés nélküli
* Végrehajtó szoftverelem: szállítási entitás
* Szállítási cím
  + port + hálózati cím+ protokoll azonosító
* Interfész: szállítási szolgáltatási primitívek

## Rajzolja fel a hálózaton közlekedő adatok beágyazását (fejrészekkel). Nevezze meg az egyes elemeket.



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



## Mire való a SOCKET primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Új kommunikációs végpont létrehozása.  
Minkét fél használja.

## Mire való a BIND primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Helyicím hozzárendelése a csatlakozóhoz.  
Ezt a szerver használja.

## Mire való a LISTEN primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Összeköttetés-elfogadási szándék bejelentése, várakozási sor hosszának megadása.  
Ezt a szerver használja.

## Mire való az ACCEPT primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Bejövő összeköttetés passzív létesítése.  
Ezt a szerver használja.

## Mire való a CONNECT primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Aktív próbálkozás összeköttetés létesítése.  
Ezt a kliens használja.

## Mire való a SEND primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Átküldés az összeköttetésen keresztül.  
Minkét fél használja.

## Mire való a RECEIVE primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Adatfogadás az összeköttetésről.  
Minkét fél használja.

## Mire való a CLOSE primitív? Melyik fél használja (szerver/kliens/mindkettő)?

Összeköttetés bontása.  
Minkét fél használja

## Mit jelent a TSAP és az NSAP? Hogy nevezzük ezeket a csatlakozókat az internet világában?

TSAP jelentése: Közlekedési szolgáltatás hozzáférési pont  
Interneten portnak nevezzük

NSAP jelentése: internet szolgáltatás hozzáférési pont  
Interneten IP-címnek nevezzük

## Honnan tudhatja egy kliens gép a szolgáltatás TSAP címét?

Állandó cím, amit egy adatbázis tárol

pl: Unix: /ect/services

Portszolgáltató (portmapper)

## Ismertesse a portszolgáltató működését.

* Speciális folyamat
* Címe ismert/állandó

1. ÖK létesítése portszolgáltatóval (PSZ)
2. Kérés küldése: hol van a szolgáltatás?
3. PSZ visszaküldi a szolgáltatás portját
4. ÖK bontása PSZ-val
5. ÖK létesítése a szolgáltatással...

## Ismertesse a kezdeti összeköttetés protokoll működését. Hogyan működik a folyamatszerver (folyamatszolgáltató)?

Folyamatszerver (process server):

* Több portot is figyel
* Ha kérés érkezik, akkor
  + elindítja a megfelelő folyamatot
  + átadja neki az ÖK-t
* Pl. unix: inetd

## Mit jelent a „jól ismert port”? Soroljon fel legalább 3 jól ismert portot.

Angolul: „Well Known Ports”  
Ezek a 1024 alatti ortok.

Jól ismert portok példái:

* FTP (TCP): 20, 21
* HTTP (TCP): 80
* HTTPS (TCP): 443

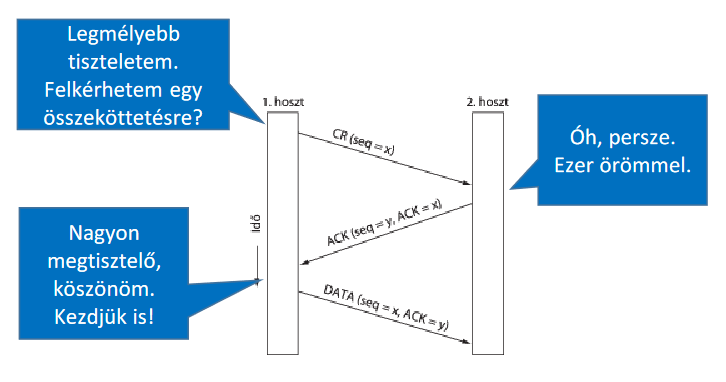
## Hogyan kezeljük a duplikált csomagok problémáját? Milyen szabályt kell betartanunk a sorszámok ismétlődésével kapcsolatban?

* A sorszámok egy tartományban mozognak, utána átfordulnak
  + Pl. 32 bites egész számok
* 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

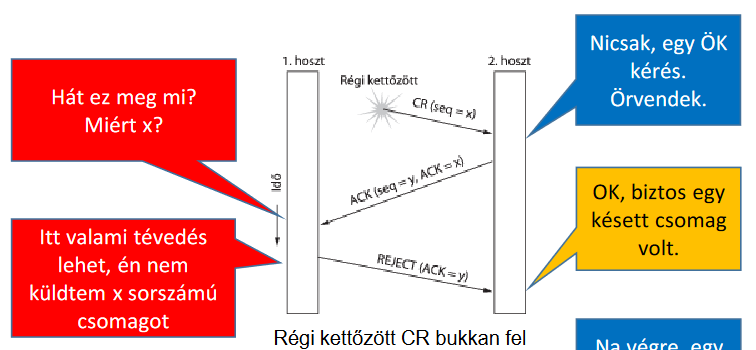
## Hogyan gondoskodunk a sokat késő csomagok eliminálásáról?

* Ugrásszámláló alkalmazása (ha túl sok ugráson át bolyong, akkor eldobjuk)
* Időcímke alkalmazása (ha túl hosszú ideig bolyong, akkor eldobjuk)
  + Bonyolult (szinkronizált órák kellenek)
  + Helyette az ugrásszámlálót használjuk

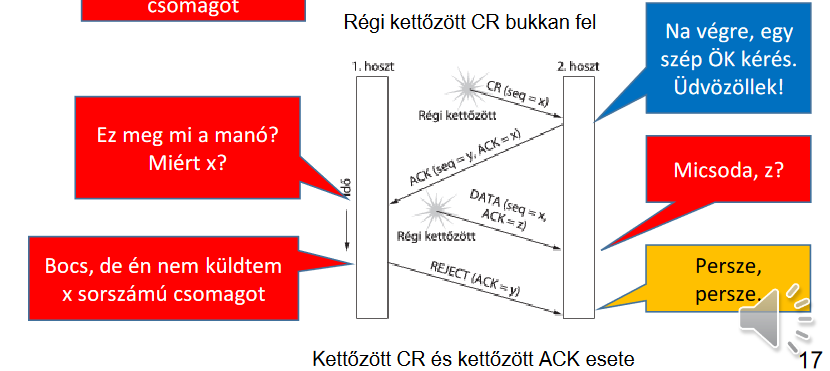
## Ismertesse a háromutas kézfogásos összeköttetés-létesítés működését.

* CR:
  + Connection request
  + Saját sorszám: x
* ACK:
  + Saját sorszám: y
  + Nyugtázza x-et
* DATA:
  + Nyugtázza y-t

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



## 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?



## Ismertesse a szimmetrikus és aszimmetrikus bontás működését. Melyik megközelítés előnyösebb és miért?

Szimmetrikus bontás

* Mindkét irányt külön kezeljük
* Mindkét irányt függetlenül bontjuk le
* Ha az egyik irányt lebontottuk, a másikban még küldhetünk adatot

Aszimmetrikus bontás

* Mint telefon esetén
* Az egyik fél bont és az ÖK megszakad
* Ez adatvesztéssel járhat

Amelyik előnyösebb: szimmetrikus bontás  
Oka: Nincs adatvesztés és csak akkor szakad meg a kapcsolat, ha mind2 fél kapcsolatot bont.

## Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldást. Hol alkalmazunk időzítőket a protokollban és miért?

A bontás elküldésekor (DR) indítunk el egy órát.  
Ezt azért tesszük, mert ha esetlegnem érkezne meg az üzenet, vagy a nyugta, akkor önkényesen tudjunk kapcsolatot bontani.

## Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben az ACK üzenet elveszik.

## Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben a válasz DR üzenet elveszik.

## Ismertesse a kapcsolat bontására alkalmazott háromutas kézfogásos megoldás működését, amennyiben a válasz DR üzenet és minden további üzenet elveszik.

## Miért szükséges a hibakezelés a szállítási rétegben?

Adatok megfelelő biztonsággal (hibátlanul) érkezzenek meg

## Mi a forgalomszabályozás feladata?

* Az adó és vevő sebességét szinkronizálja
* Egy gyors adó ne töltsön túl egy lassú vevőt

## Ismertesse a csúszóablakos protokoll működését.

* Egyesíti a korábbi eszközöket, de
* változó méretű ablakokat (pufferméret) használ
* Különválasztja 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

## 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?

6

## 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)?

14, 15

## Mit jelent a nyalábolás a szállítási rétegben?

* Több szállítási összeköttetés ugyanazt a hálózati címet használja
* Több TSAP -> egy NSAP
* Ez a tipikus, hiszen általában 1 hálózati cím tartozik egy géphez
* Pl. a TCP ezt használja

## Mit jelent a fordított nyalábolás a szállítási rétegben?

* Egy szállítási összeköttetés több hálózati címet (interfészt) is használ
* Egy TSAP -> több NSAP
* Ok: sávszélesség megnő
* k db. hálózatai összeköttetés: k-szoros sávszélesség
* Pl. SCTP (Stream Control Transmission Protocol) is ilyen megoldást használ

## Mi a torlódáskezelés feladata?

A torlódás az útválasztókban jön létre, de a végpontokon kezeljük.

## Ismertesse a forgalomszabályozás és a torlódáskezelés feladatait. Mi a különbség a két fogalom között?

Forgalomszabályozás

* Adó és vevő sebességének illesztése
* Kezelés:
  + Vevő információt küld az állapotáról (ablakméret)
  + Adó ennek megfelelően ad (elküldött csomagok száma)
  + Csúszóablakos protokoll

Torlódáskezelés (congestion control)

* Ha
  + túl sok gép
  + túl gyorsan
  + túl sok
* csomagot küld a hálózatba, akkor torlódás keletkezik.
* A torlódás az útválasztókban jön létre, de a végpontokon kezeljük.
* Kezelés:
  + Érzékeljük a torlódást
  + Ennek megfelelően a hálózatba küldött csomagok számának szabályozása

A különbség, hogy míg a forgalomszabályozásnál az adó és vevő sebességét egymásra illesszük, míg a forgalomszabályozásnál a küldött csomagok számát szabályozzuk.

## Milyen módon avatkozhatunk be torlódás érzékelése esetén?

Az átviteli sebességet szabályozzuk a hálózathoz az adó oldalon.

## Mit jelent a torlódás? Milyen látható jelei vannak a hálózatban?

A hálózatban kevesebb adat folyik át a csatornán, mint amennyit küldünk.  
Látható jelei: A hálózati csatornában az adatok elkezdenek egyre jobban felgyülemleni egy adott ponttól, ami akár a hálózat túlterhelését is előidézheti..

## Rajzolja fel a hasznos átbocsátást a felajánlott forgalom függvényében. Mutassa meg a torlódás hatását.

Hatása: Goodput lassul, majd visszaesik

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

Hatása: Késleltetés nő

## Mit jelent, hogy a torlódáskezelés hatékony?

A teljes rendelkezésre álló sávszélességet ki tudja használni

## Mit jelent, hogy a torlódáskezelés igazságos?

Sávszélesség megfelelő elosztása a hálózatot használóentitások között

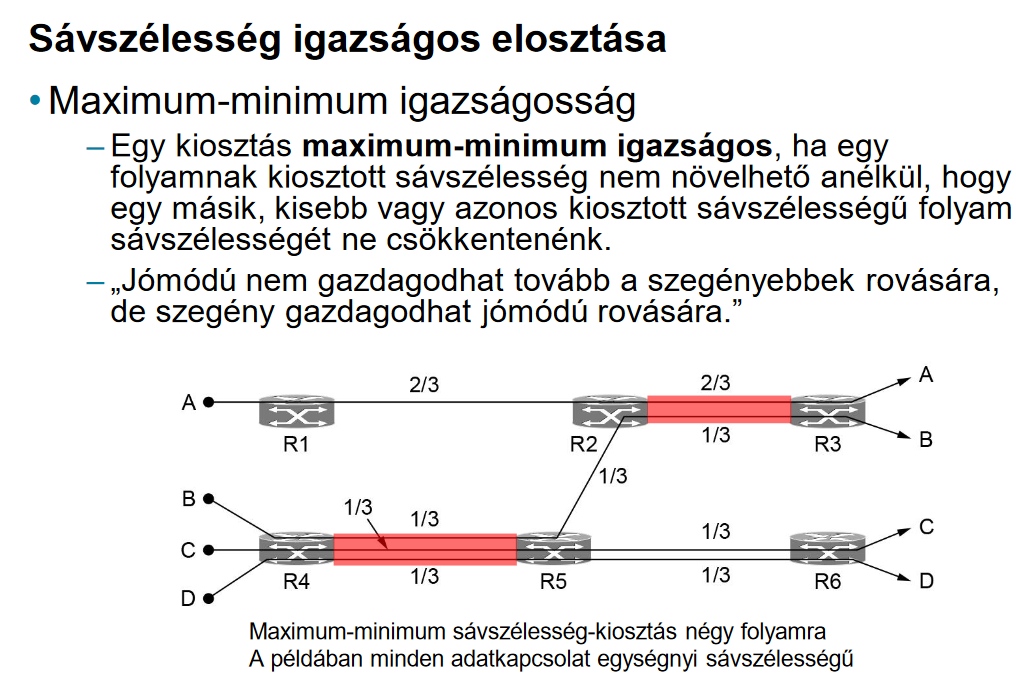
## Mit jelent, hogy a torlódáskezelés konvergens?

Gyorsan képes követni a forgalmi igényeket

## Definiálja a maximum-minimum igazságosság elvét.

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.

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



## Mit jelent az explicit torlódásérzékelés? Adjon rá példát.

A torlódásról egyértelmű jelzést adunk.  
pl: ECN

## Mit jelent az implicit torlódásérzékelés? Adjon rá példát.

A torlódást egyéb jelekből „sejtjük”  
(pl. körülfordulási idő növekedése)

## Mit jelent, hogy egy torlódásérzékelésre használt jelzés pontos vagy 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)

* Ha van torlódás: sebesség csökkentése
* Ha nincs torlódás: sebesség növelése

## Nevezzen módszereket, amelyekkel a torlódás érzékelhető.

* Explicit
* Implicit
* Pontos
* Pontatlan

## Milyen alapvető szabályozási törvényt alkalmazunk a torlódáskezelés során?

* Ha van torlódás: sebesség csökkentése
* Ha nincs torlódás: sebesség növelése

## Ismertesse az AIMD szabályozási törvényt. Illusztrálja ennek működését két állomás esetén.

* Ha van torlódás: sebesség csökkentése
* Ha nincs torlódás: sebesség növelése

## Miért nem okoz gondot a vezeték nélküli hálózatokban gyakori csomagvesztés a torlódás érzékelésében (akkor sem, ha az érzékelés csomagveszteségen alapul)?

* A vezeték nélküli adatkapcsolati réteg érzékeli a keretvesztést
  + Azonnal javít: újraküld néhány μs-on belül
* Az átviteli hiba gyorsan javításra kerül, a szállítási réteg nem veszi észre
  + A szállítási rétegben az időzítők a ms ... s nagyságrendben vannak

## Foglalja össze röviden a TCP által nyújtott szolgáltatásokat.

* Nagyon sok új szolgáltatás:

Összeköttetés alapú

A bájtok megbízható módon, sorrendben, 1x kerülnek átvitelre

Tetszőleges hosszúságú lehet

Forgalomszabályozás hangolja az adót a vevőhöz

Torlódáskezelés hangolja az adót a hálózathoz

## Foglalja össze röviden az UDP által nyújtott szolgáltatásokat.

* Kb. annyit tud mint egy IP csomag

Összeköttetés nélküli (datagramm)

Üzenet elveszhet, duplikálódhat, sorrendjük keveredhet

Korlátos hosszúságú üzenet

Adó adhat a vevő állapotától függetlenül

Adó adhat a hálózat állapotától függetlenül

• (UDP: User Datagram Protocol)

• UDP felépítése és használata

• Alkalmazási példák:

– Távoli eljáráshívás

– Valós idejű szállítási protokollok

▪ RTP

▪ RTCP

▪ Lejátszás puffereléssel és jitter-szabályozással

## Hasonlítsa össze az UDP és TCP tulajdonságait.

A képen asztal látható

Automatikusan generált leírás

## Magyarázza el, hogy mire szolgálnak az UDP fejrészben a forrásport és célport mezők.

• Forrásport, célport – Portok azonosítói

A képen asztal látható

Automatikusan generált leírás

## Mit jelent az IP fejrész és mire szolgál? Mi a kapcsolata az UDP ellenőrző összeggel?

?? csak IP álfejrészt találtam és nem 100% hogy oda az kell

## Mi a socket rendszerhívás célja?

Csatlakozó Választás

## Mire szolgál a bind rendszerhívás?

?? A 2. hoszt levelezőszervere a 25-ös TSAP-hoz csatlakozik

helyi cím hozzárendelése az csatlakozóhoz

## Mire szolgál a sendto és recvfrom rendszerhívás?

??nincs benne a pptbe

## Mit jelent, hogy a revfrom blokkoló hívás?

## ??

## Mi a távoli eljáráshívás?

Úgy viselkedjen, mintha hagyományos (helyi) függvényhívás lenne

## Mit a jelent a csonk (stub)?

Úgy viselkedik, mint egy helyi függvényhívás

– Elvégzi a hálózati feladatokat:

▪ Paramétereket üzenetbe pakolja (marshaling)

▪ Elküldi az üzenetet

▪ Veszi a választ

▪ Visszatér az eredménnyel a hívóhoz

## Magyarázza el a cliens-oldali és  aszerver-oldali csonkok szerepét és működését.

A képen szöveg látható

Automatikusan generált leírás

## Mire használjuk az RTP protokollt?

: Real Time Transfer Protokol

– 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?

RTCP: Real-Time Transport Control Protocol

• Mintákat nem szállít

• Feladata:

1. Visszacsatolást kezeli:

▪ Hálózat tulajdonságairól a szervernek (késleltetés, jitter, sávszélesség, torlódás)

▪ Ez alapján képes folyamatosan állítani a minőséget (pl. kódolás, felbontás)

1. Szinkronizációt biztosít

▪ A különféle folyamok óráit egymáshoz szinkronizálja

1. 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)

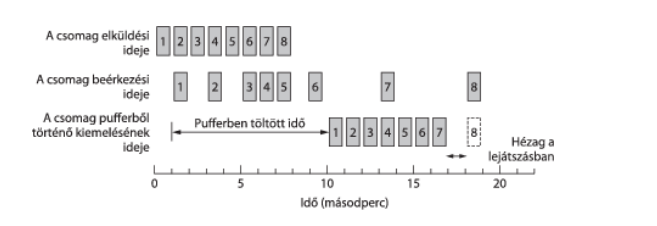
▪ Egyes alkalmazásokban kis értéken kell tartani (pl. telefon)

– Késleltetés ingadozása (jitter)

▪ Nagyon zavaró! Kompenzálni kell a vevőben!

## Hogyan kompenzáljuk a jitter hatását a vevőben?

Puffereljük



## Mit jelent a lejátszási pont?

Lejátszási pont (playback point):

– Mennyit kell várnia vevőnek a lejátszás indításáig

– Pl. a csomagok 99%-a megérkezzen

## Hogyan azonosítunk egy csatlakozót (címzés)?

IP cím alapján

Portok száma alapján

Protokoll alapján

## Mit jelent a „jól ismert port”?

1024 alatti portok

## Magyarázza el az inted szerepét és működését.

inetd figyel több portot is

▪ szükség esetén elindítja a megfelelő démont

▪ nem kell egyszerre sok démonnak feleslegesen futnia

## Milyen portszámokat kapnak a kliens folyamatok?

A képen asztal látható

Automatikusan generált leírás

## Kitől kapják a portszámokat?

Általában ideiglenes portokra csatlakoznak

– Az operációs rendszer választja

– Véletlenszerű

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

Nincs adatszórás vagy többesküldés

## Mit jelent, hogy a TCP egy bájtfolyamat biztosít?

Nincs üzenet, nincs üzenethatár

– A szegmensek kontroll infót is szállítanak (pl. ACK) →piggyback

## Mikor küldi el a TCP a rábízott adatokat?

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)

## Hogyan lehet gyors továbbítást kikényszeríteni?

## A TCP milyen adategységet sorszámoz?

minden bájt rendelkezik egy 32bites sorszámmal

## Mi korlátozza TCP szegmens méretét?

IP adatmező

▪ MTU (legnagyobb átvihető adategység)

–Pl. Ethernet: 1500B

–Megoldások:

• Darabolás a hálózatban

• rontja a teljesítőképességet

• már nem gyakran használt

• Útvonal MTU meghatározás

## 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)

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

Ablakméret (WIN) – A vevőben rendelkezésre álló puffer mérete

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

CWR, ECE: Torlódás jelzése

▪ 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, RTS, SYN és FIN bitek szerepét.

• ACK: Jelzi a nyugtaszám érvényességét

• PSH (PUSH): késedelem nélküli továbbítás kérése

– Ne legyen pufferelés

• RST (RESET): Összeköttetés helyreállítás (valami baj történt)

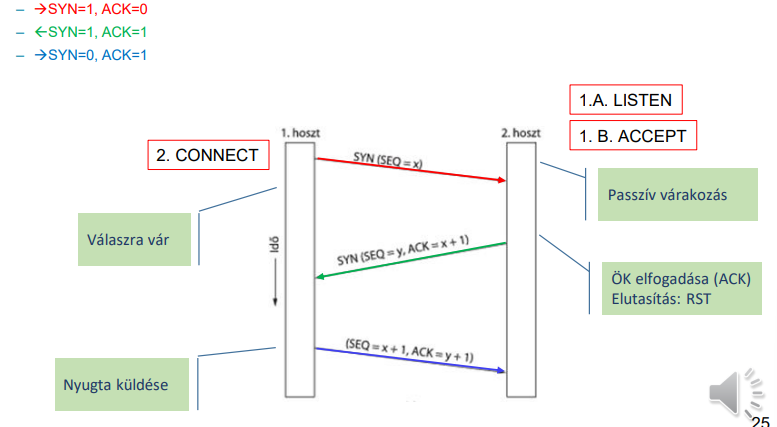
• SYN: kapcsolat kiépítés

– Jelzi a CONNECTION REQUEST és CONNECTION ACCEPTED üzeneteket

• FIN: kapcsolat bontása

– Jelzi, hogy a küldőnek nincs több adata

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

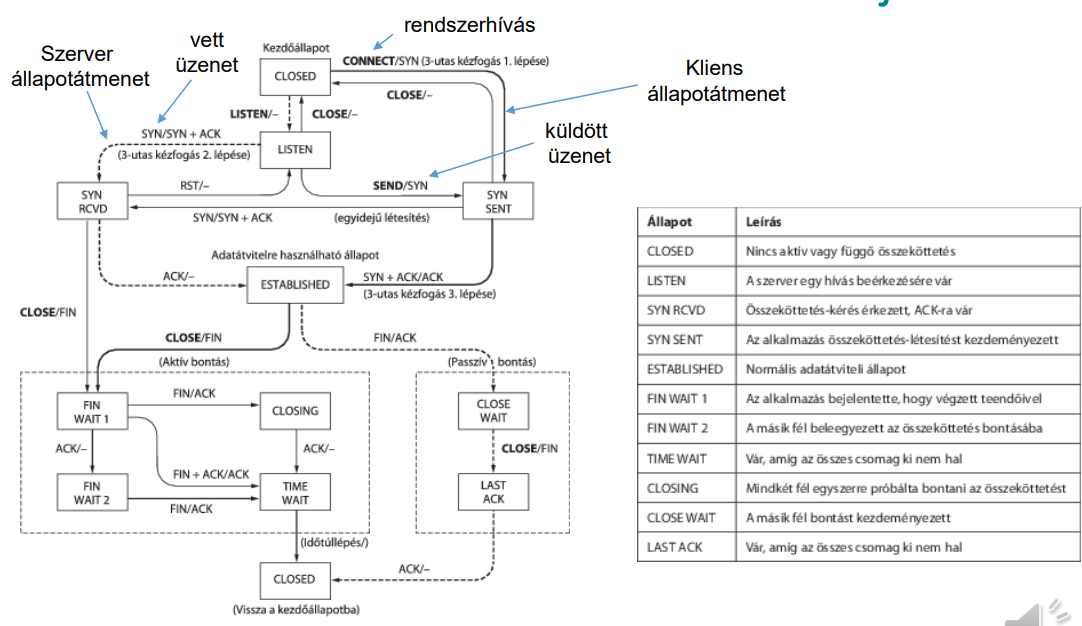


## Ismertesse a TCP összeköttetés lebontásának módját.

A képen szöveg látható

Automatikusan generált leírás

## A 27. oldalon található állapotgép segítségével magyarázza el a TCP összeköttetés felépítésének menetét.



## A 27. oldalon található állapotgép segítségével magyarázza el a TCP összeköttetés lebontásnak menetét.

## 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

= 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:

a.)    ACK=67123, WIN=512

b.)    ACK=512, WIN=0

## Egy szervernek 2kB üres pufferterület áll rendelkezésre egy TCP összeköttetés kezelésére.

## Most a következő (512 adatbájtot tartalmazó) üzenet érkezik a szerverhez:

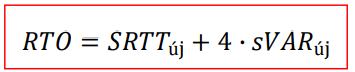
               ADAT: 512B, SEQ=3000

               Mi lesz a szerver által küldött következő üzenetben az ACK és a WIN értéke? Miért?

## Hogyan lehet a szegmens elveszését időzítő segítségével detektálni?

? 3 nyugtás szabállyal detetektálni.

## Hogyan célszerű beállítani az időzítő értékét, ha ismerjük az átlagos késleltetési időt (SRTT) és a késleltetési idő szórását (SVAR)?

****

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

A képen szöveg látható

Automatikusan generált leírás

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

A képen szöveg látható

Automatikusan generált leírás

## Mitől jöhet létre torlódás? Mit jelent a torlódási ablak?

Ha a hálózati terhelés túl nagy

– Csomagok feltorlódnak az útválasztókban

– Csomagok késnek, elvesznek

## Hogyan használja a TCP a forgalomszabályozási és a torlódási ablakot?

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

• Ez a nyugtaórajel (ack clock)

• Az adó ennek megfelelő sebességgel ad

– Elkerüli a felesleges sorbanállást az útvonalválasztókon

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

Kezdeti munkapont elérése AI val lassú

helyette indításkor exponenciálisan növekvő ablakméretet használunk  
ez a lassú kezdés

– Lassú kezdés működése:

▪ Minden nyugta vételénél:

–ablakméret eggyel nő

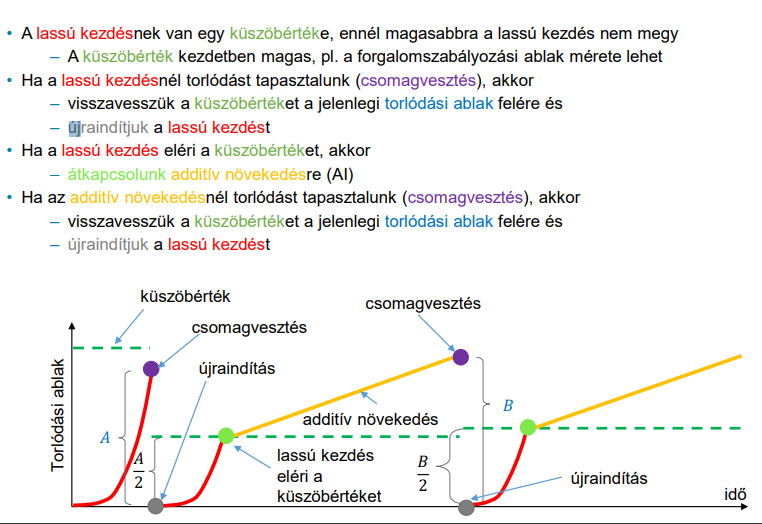
–új csomagot küld a régi helyett

új csomagot küld az új üres helyre

▪ Úton lévő csomagok száma:

–RTT alatt duplázódik

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



## Mit értünk „gyors újraküldésen” a TCP Tahoe protokollban?

• Torlódás/csomagvesztés érzékelése:

• Időtúllépés vagy

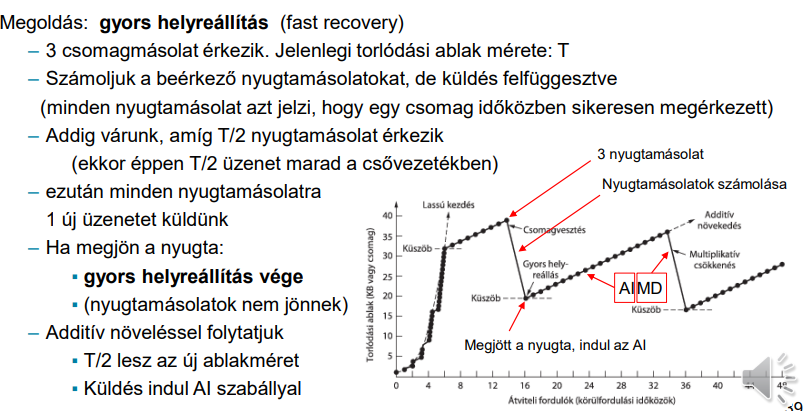
• 3 nyugtamásolat

• ekkor a hiányzó csomagokat azonnal újraküldjük

• Ez a gyors újraküldés (fast retransmission)

## 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?

AIMD:

• torlódási ablak additív növelése (+1), amíg csak lehet

• multiplikatív csökkentése torlódás esetén (/2)

## Magyarázza el a szelektív nyugtázás működését.

Szelektív nyugtázás (SACK)

– Kummulatív (halmozott) ACK mindig van

– Szelektív ACK opcionális (széles körben támogatott)

## Magyarázza el az explicit torlódásjelzés működését az ECE és CWR bitek segítségével.

• Explicit torlódásjelzés (ECN – Explicit Congestion Notification)

– IP útválasztó a csomagban beállítja a torlódás bitet (lásd Differenciált szolgáltatások)

– A vevő a válasz TCP fejrészben beállítja az ECE bitet (ECN Echo)

– A küldő az ECE hatására úgy reagál, mintha csomagvesztés lenne

– A küldő CWR bitet beállítja a következő csomagban (Congestion Window Reduced)