

# Számítógépes Hálózatok

## 6. Előadás: Adatkapcsolati réteg MAC alréteg

Based on slides from **Zoltán Ács ELTE** and D. Choffnes Northeastern U., Philippa Gill from StonyBrook University , Revised Spring 2016 by S. Laki

# Csúszó-ablak protokollok 1 / 2

## ALAPOK (ÁLTALÁNOS)

- Egy adott időpontban egyszerre több keret is átviteli állapotban lehet.
- A fogadó  $n$  keretnek megfelelő méretű puffert allokál.
- A küldőnek legfeljebb  $n$ , azaz ablak méretnyi, nyugtázatlan keretet küldése engedélyezett.
- A keret sorozatbeli pozíciója adja a keret címkéjét. (sorozatszám)

## ALAPOK (FOGADÓ)

- A keret nyugtázója tartalmazza a következőnek várt keret sorozatszámát.
  - ▣ *kumulatív nyugta* – Olyan nyugta, amely több keretet nyugtáz egyszerre. Például, ha a 2,3 és 4 kereteket is fogadnánk, akkor a nyugtát 5 sorszám tartalommal küldenénk, amely nyugtázza mind a három keretet.
- A hibás kereteket el kell dobni.
- A nem megengedett sorozatszámmal érkező kereteket el kell dobni.

# Csúszó-ablak protokollok 2/2

## JELLEMZŐK (ÁLTALÁNOS)

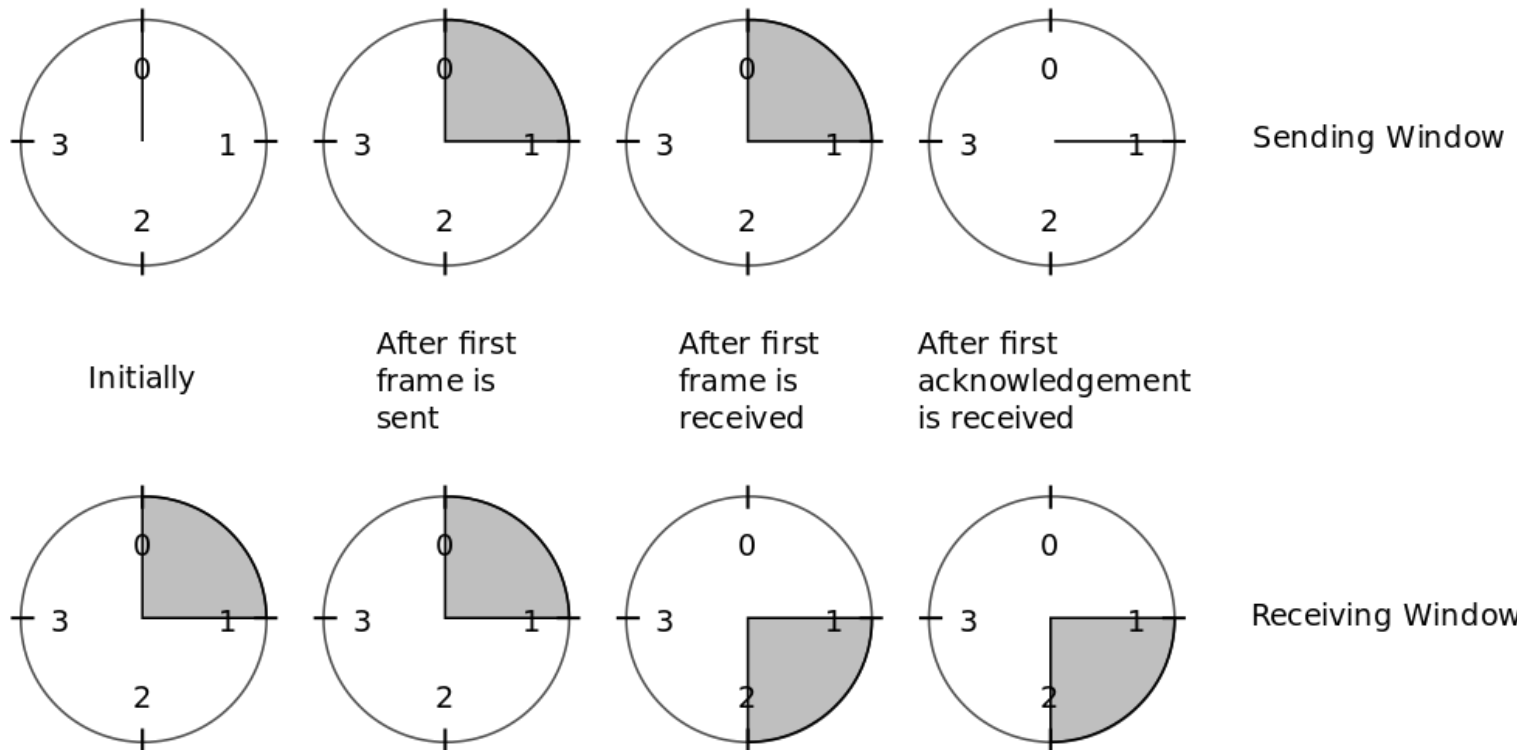
- A küldő nyilvántartja a küldhető sorozatszámok halmazát. (*adási ablak*)
- A fogadó nyilvántartja a fogadható sorozatszámok halmazát. (*vételi ablak*)
- A sorozatszámok halmaza minden esetben véges.
  - ▣  $K$  bites mező esetén:  $[0..2^K - 1]$ .
- A adási ablak minden küldéssel szűkül, illetve nő egy nyugta érkezésével.

## JELLEMZŐK (GYAKORLATI ALKALMAZÁS ESETÉN)

- gyakorlatban kétirányú adatfolyamot kell kezelni (*duplex csatorna*)
  - ▣ két különböző szimplex csatorna használata (*két áramkör használata*)
  - ▣ egy csatorna használata (*egy áramkör használata*)
    - **piggybacking módszer**– a kimenő nyugtákat késleltetjük, hogy rá tudjuk akasztani a következő kimenő adatkeretre (*ack mező használata*);

# Csúszó ablak

4



A sliding window with a 2-bit sequence, of size 1



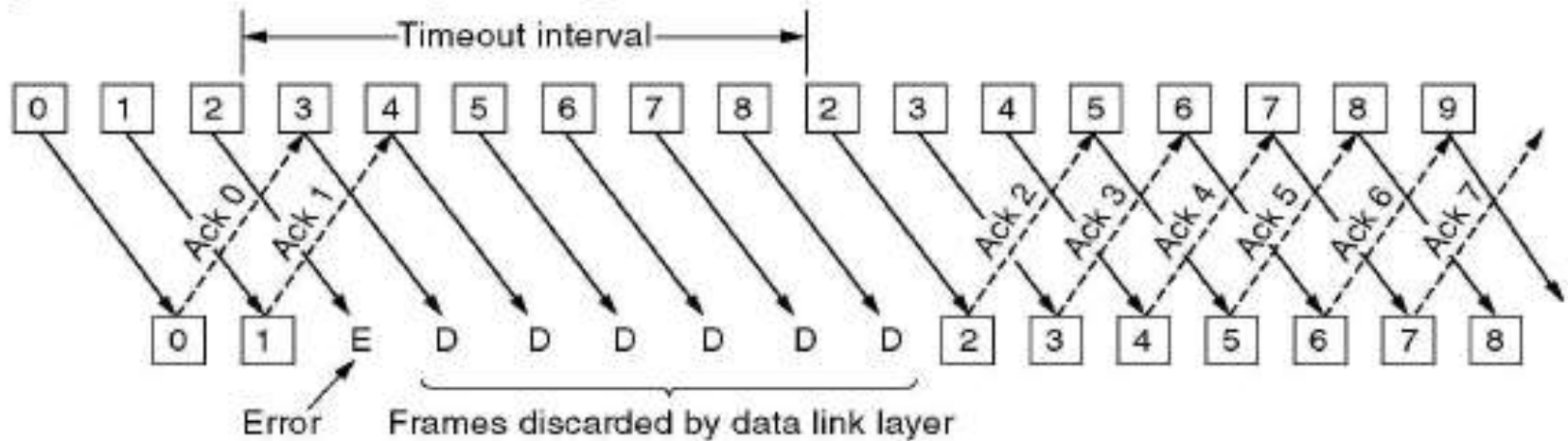
# Pipelining

- Eddig feltételeztük, hogy *a keret vevőhöz való megérkezéséhez és a nyugta visszaérkezéséhez együttesen szükséges idő elhanyagolható.*
  - ▣ a nagy RTT a sávszélesség kihasználtságra hatással lehet
  - ▣ **Ötlet:** egyszerre több keret küldése
  - ▣ Ha az adatsebesség és az RTT szorzata nagy, akkor érdemes nagyméretű adási ablakot használni. (*pipelining*)
- Mi van ha egy hosszú folyam közepén történik egy keret hiba?
  1. „visszalépés N-nel”, avagy angolul *go-back-n*
  2. „szelektív ismétlés”, avagy angolul *selective-repeat*

# Go-Back-N

- A sliding window protocol where
  - ▣ the receiver's window size is fixed to 1,
  - ▣ while the sender has window size  $> 0$ .
  
- After receiving a damaged frame
  - ▣ Receiver discards all subsequent frames
  - ▣ Sender retransmits the damaged frame and all its successors after the times out

# Go-Back-N

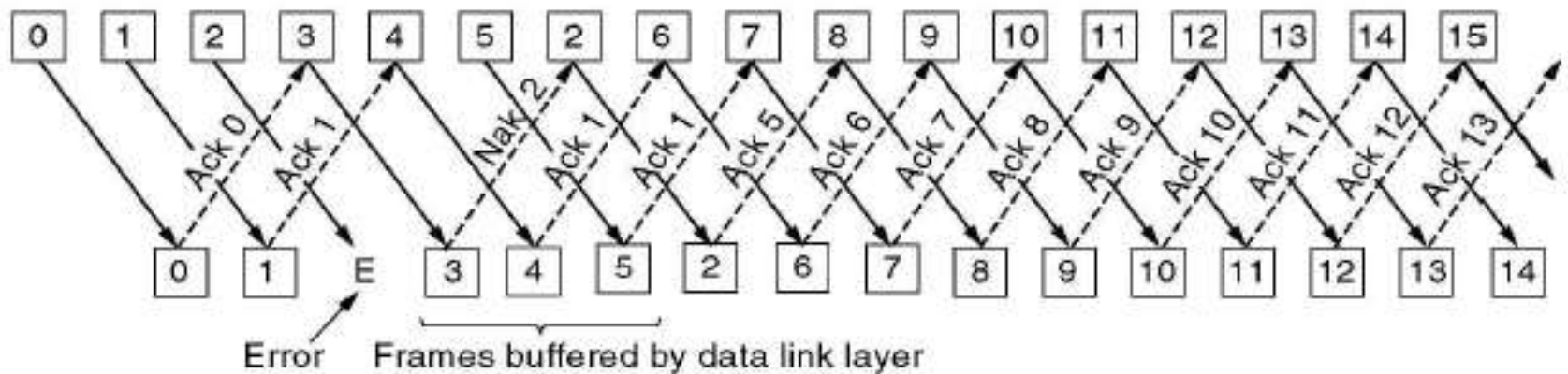




# Selective Repeat

- Receiver's window size is  $n$  (  $n > 1$  )
  - ▣ At most  $n$  frames can be buffered
- Receiver stores all the correct frames following the bad one
- The sender retransmits only the bad frame not all its successors

# Selective Repeat



# Ethernet keret

802.3 Ethernet frame structure

| Preamble | Start of frame delimiter | MAC destination | MAC source | 802.1Q tag (optional) | Ethertype (Ethernet II) or length (IEEE 802.3) | Payload                             | Frame check sequence (32-bit CRC) | Interframe gap |
|----------|--------------------------|-----------------|------------|-----------------------|--|-------------------------------------|-----------------------------------|----------------|
| 7 octets | 1 octet                  | 6 octets        | 6 octets   | (4 octets)            | 2 octets                                       | 42 <sup>[note 2]</sup> –1500 octets | 4 octets                          | 12 octets      |
|          |                          | 64–1522 octets  |            |                       |  |                                     |                                   |                |
|          |                          | 72–1530 octets  |            |                       |  |                                     |                                   |                |
|          |                          | 84–1542 octets  |            |                       |  |                                     |                                   |                |

# Közeg hozzáférés vezérlése

## Media Access Control (MAC)

# Mi az a közeg hozzáférés ?

13

- ❑ Ethernet és a Wifi is többszörös hozzáférést biztosító technológiák
  - ▣ Az átviteli közegen több résztvevő osztozik
    - Adatszórás (broadcasting)
  - ▣ Az egyidejű átvitel **ütközést** okoz
    - Lényegében megghiúsítja az átvitelt
- ❑ Követelmények a Media Access Control (MAC) protokolljaival szemben
  - ▣ Szabályok a közeg megosztására
  - ▣ Stratégiák az ütközések detektálásához, elkerüléséhez és feloldásához

# MAC alréteg

14

- Eddigi tárgyalásaink során pont-pont összeköttetést feltételeztünk.
- Most az adatszóró csatornát (angolul *broadcast channel*) használó hálózatok tárgykörével foglalkozunk majd.
  - ▣ **Kulcskérdés:** *Melyik állomás kapja a csatornahasználat jogát?*
- A csatorna kiosztás történhet:
  1. statikus módon (FDM, TDM)
  2. dinamikus módon
    - a) verseny vagy ütközés alapú protokollok (ALOHA, CSMA, CSMA/CD)
    - b) verseny-mentes protokollok (bittérkép-alapú protokollok, bináris visszaszámlálás)
    - c) korlátozott verseny protokollok (adaptív fa protokollok)

# Statikus csatornakiosztás

15

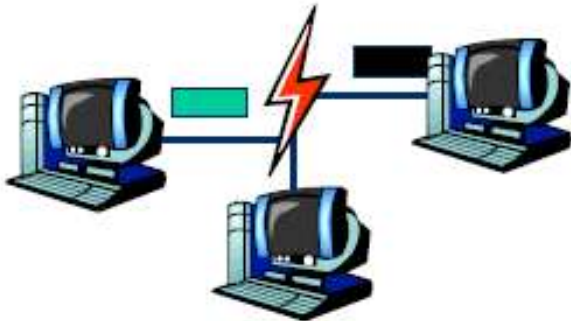
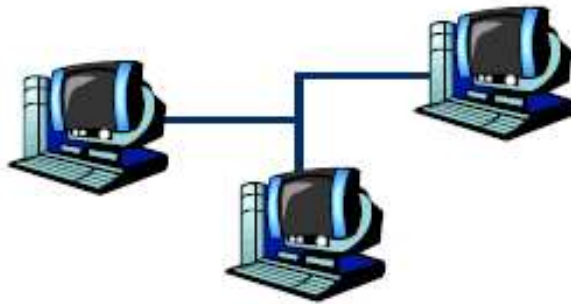
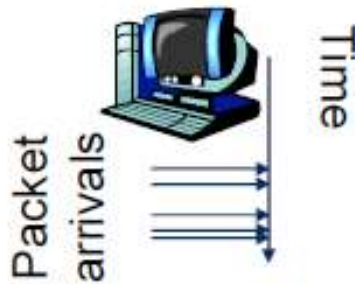
## Frekvenciaosztásos nyalábolás

- $N$  darab felhasználót feltételezünk, a sávszélet  $N$  egyenlő méretű sávra osztják, és minden egyes sávhoz hozzárendelnek egy felhasználót.
- Következésképpen az állomások nem fogják egymást zavarni.
- Előnyös a használata, ha fix számú felhasználó van és a felhasználók nagy forgalmi igényt támasztanak.
- Löketszerű forgalom esetén használata problémás.

## Időosztásos nyalábolás

- $N$  darab felhasználót feltételezünk, az időegységet  $N$  egyenlő méretű időrésre – úgynevezett *slot*-ra – osztják, és minden egyes részhez hozzárendelnek egy felhasználót.
- Löketszerű forgalom esetén használata nem hatékony.

# Dinamikus csatornakiosztás



## 1. Állomás modell

- ▣ N terminál/állomás
- ▣ Annak a valószínűsége, hogy  $\Delta t$  idő alatt csomag érkezik  $\lambda \Delta t$ , ahol  $\lambda$  az érkezési folyamat rátája.

## 2. Egyetlen csatorna feltételezés

- ▣ Minden állomás egyenrangú.
- ▣ Minden kommunikáció egyazon csatornán zajlik.
- ▣ Minden állomás tud ezen küldeni és fogadni csomagot.

## 3. Ütközés feltételezés

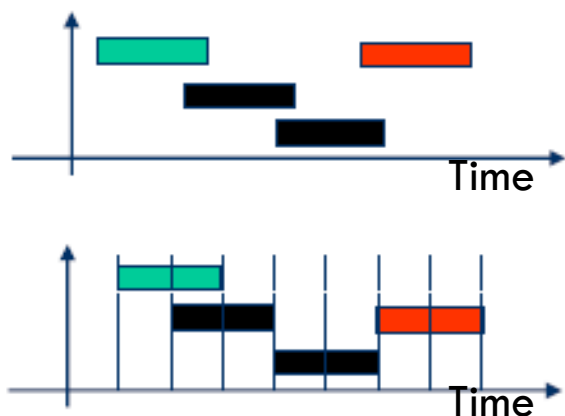
- ▣ Ha két keret egy időben kerül átvitelre, akkor átlapolódnak, és az eredményül kapott jel értelmezhetetlenné válik.
- ▣ Ezt nevezzük ütközésnek.

## 4. Folytonos időmodell VS diszkrét időmodell

## 5. Vivőjel értékelés VS nincs vivőjel érzékelés



# Dinamikus csatornakiosztás



## Használt időmodell

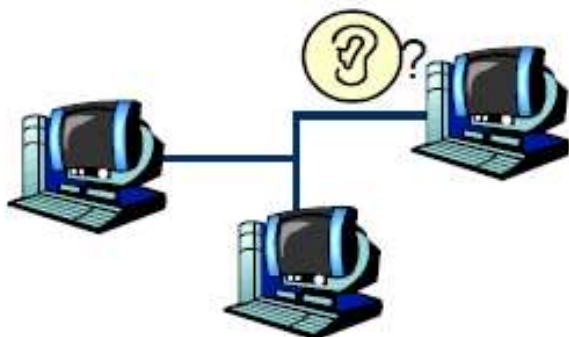
Kétféle időmodellt különböztetünk meg:

- a) **Folytonos** – Mindegyik állomás tetszőleges időpontban megkezdheti a küldésre kész keretének sugárzását.
- b) **Diszkrét** – Az időt diszkrét résekre osztjuk. Keret továbbítás csak időrés elején lehetséges. Az időrés lehet *üres*, *sikeres* vagy *ütközéses*.

## Vivőjel érzékelési képesség

Az egyes állomások vagy rendelkeznek ezzel a tulajdonsággal vagy nem.

- a) Ha **nincs**, akkor az állomások nem tudják megvizsgálni a közös csatorna állapotát, ezért egyszerűen elkezdnek küldeni, ha van rá lehetőségük.
- b) Ha **van**, akkor állomások meg tudják vizsgálni a közös csatorna állapotát a küldés előtt. A csatorna lehet: foglalt vagy szabad. Ha a foglalt a csatorna, akkor nem próbálják használni az állomások, amíg fel nem szabadul.

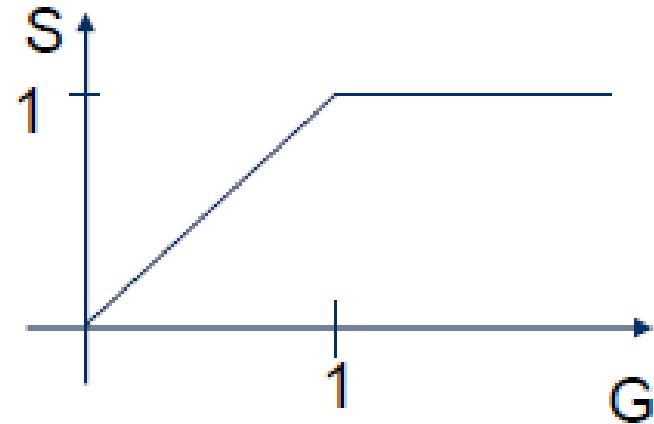


Megjegyzés: Ez egy egyszerűsített modell!

# Átvitel és terhelés

## □ Terhelés ( $G$ )

- A protokoll által kezelendő csomagok száma egy időegység alatt (beérkező kérések)
- $G > 1$ : túlterhelés
- A csatorna egy kérést tud elvezetni

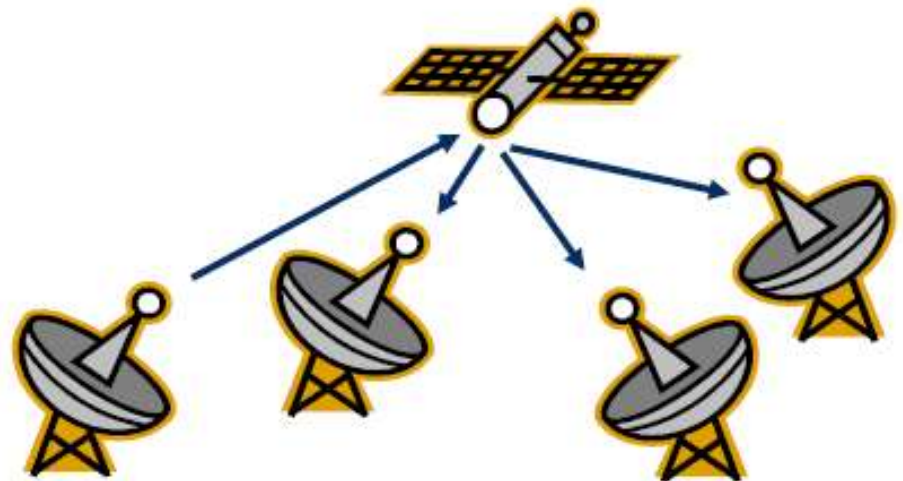
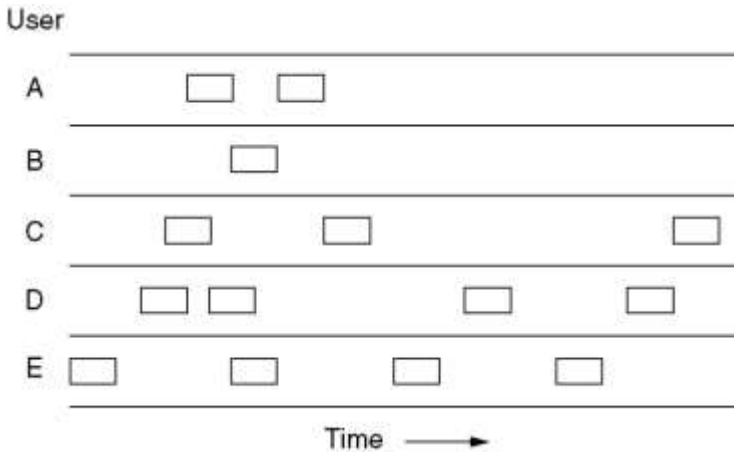


## □ Ideális esetben

- Ha  $G < 1$ ,  $S = G$
- Ha  $G \geq 1$ ,  $S = 1$
- Ahol egy csomag kiküldése egy időegységet vesz igénybe.

© 2015 Pearson Education, Inc. or its affiliate(s). All rights reserved. Pearson Education, Inc., publishing as Pearson Benjamin Cummings, 101 Philip Drive, Assinippi Park, New York, NY 10984-2135

- Alacsony költségű, nagyon egyszerű megoldás



# ALOHA

20

- ❑ Topológia: broadcast rádió több állomással
  - ❑ Protokoll:
    - Az állomások azonnal küldenek
    - A fogadók minden csomagot nyugtáznak
    - Nincs nyugta = ütközés, véletlen ideig vár, majd újraküld
- Egyszerű, de radikális megoldás
  - Korábbi megoldások, mind felosztották a csatornát
    - TDMA, FDMA, etc.
  - Kevés küldő esetére készült

# Teljesítmény elemzés -Poisson Folyam

- A „**véletlen érkezések**” egyik ünnepelt modellje a sorban-állás elméletben a Poisson folyamat.
- A modell feltételezései:
  - ▣ Egy érkezés valószínűsége egy rövid  $\Delta t$  intervallum alatt arányos az intervallum hosszával és nem függ az intervallum kezdetétől (ezt nevezzük **memória nélküli** tulajdonságnak)
  - ▣ Annak a valószínűsége, hogy több érkezés történik egy rövid  $\Delta t$  intervallum alatt közelít a nullához.

# Teljesítmény elemzés –Poisson eloszlás

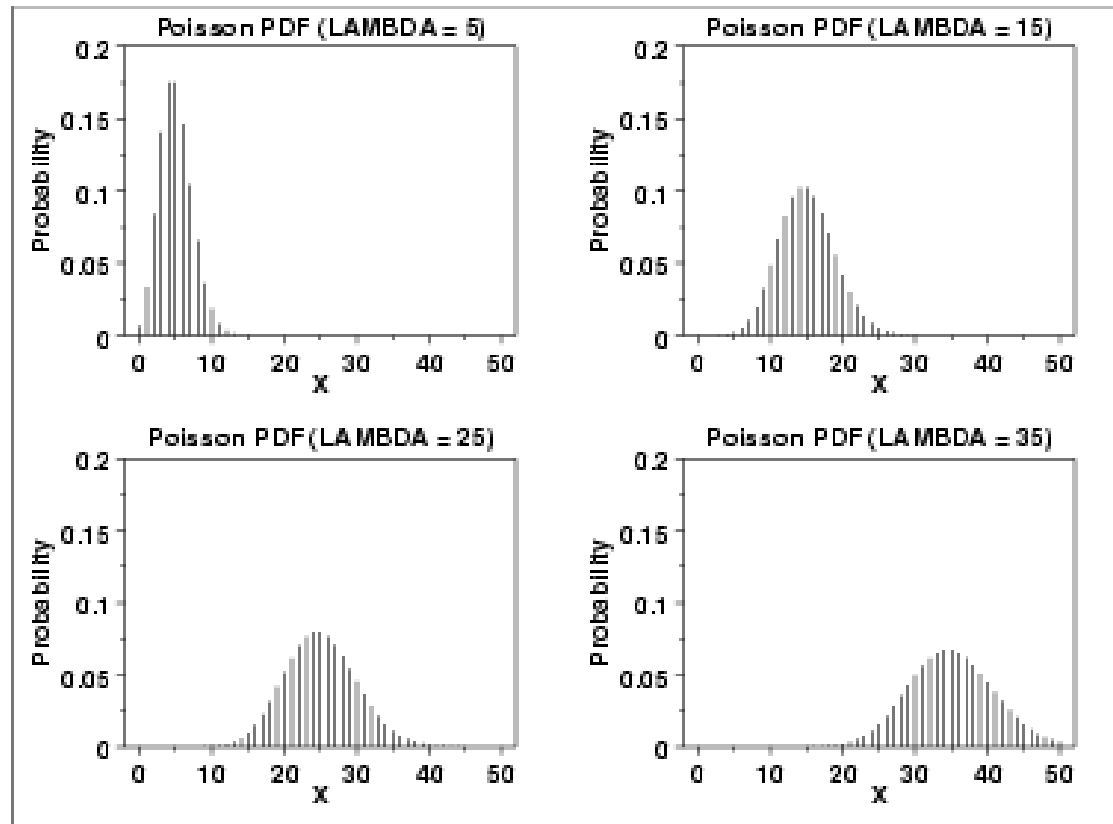
Annak a valószínűsége, hogy  $k$  érkezés történik egy  $t$  hosszú intervallum során:

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

ahol  $\lambda$  az érkezési ráta. Azaz ez egy egy-paraméteres modell, ahol csak  $\lambda$ -át kell ismernünk.

# Poisson Eloszlás példák

23



# ALOHA vizsgálata

## □ Jelölés:

- $T_f$  = keret-idő (feldolgozási, átviteli és propagációs)
- $S$ : A sikeres keret átvitelek átlagos száma  $T_f$  idő alatt; (*throughput*)
- $G$ :  $T_f$  idő alatti összes átviteli kísérletek átlagos száma
- $D$ : Egy keret küldésre kész állapota és a sikeres átvitele között eltelt átlagos idő

## □ Feltételezéseink

- Minden keret konstans/azonos méretű
- A csatorna zajmentes, hibák csak ütközések miatt történnek
- A keretek nem kerülnek sorokba az egyedi állomásokon
- Egy csatorna egy Poisson folyamként viselkedik



# ALOHA vizsgálata

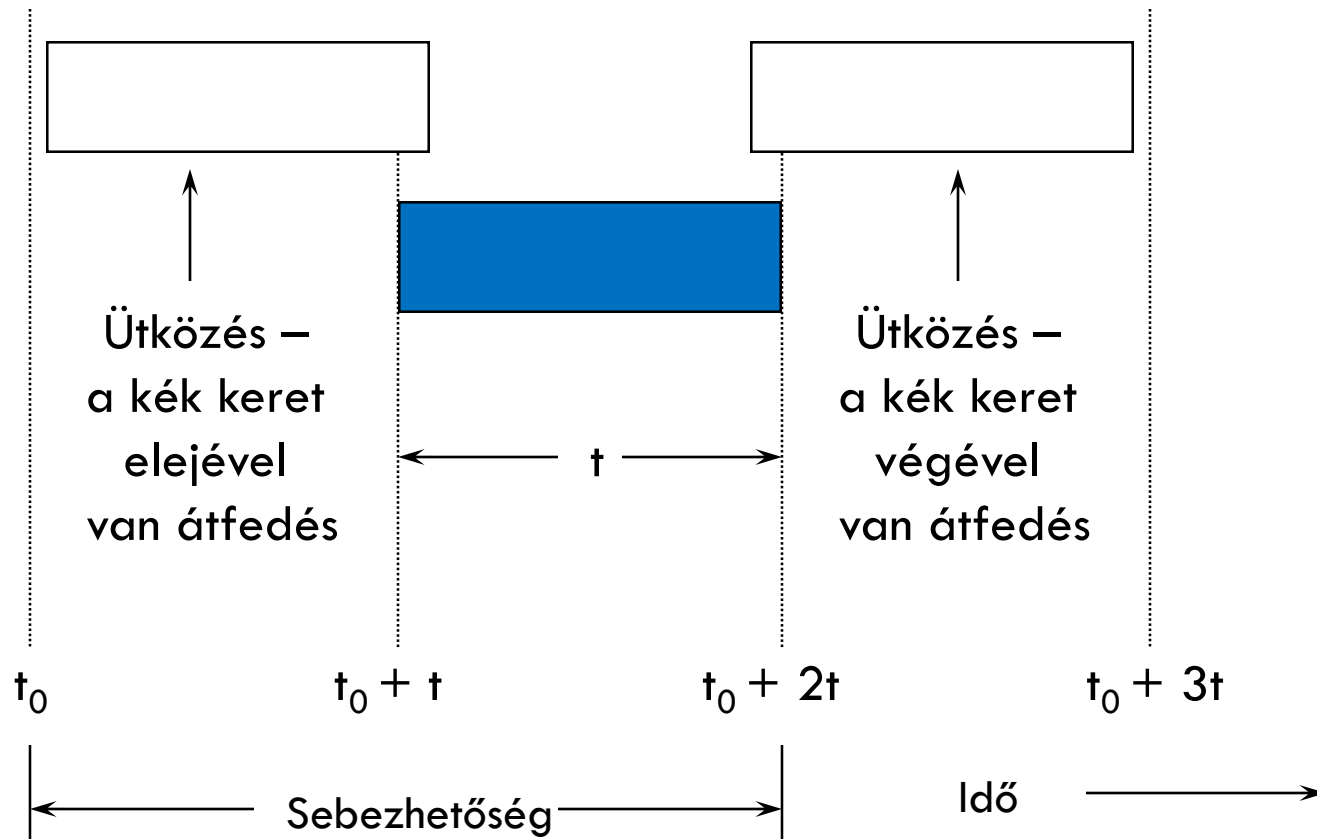
- Mivel  $S$  jelöli a „jó” átviteleket egy keret idő alatt és  $G$  jelöli az összes átviteli kísérletet egy keret idő alatt, így a következő összefüggést írhatjuk:

$$S = S(G) = G \times (\text{A „jó” átvitelek valószínűsége})$$

- A sebezhetőségi idő egy keret sikeres átviteléhez:  $2T_f$
- Azaz a „jó” átvitel valószínűsége megegyezik annak a valószínűségével, hogy a sebezhetőségi idő alatt **nincs** beérkező keret.

# ALOHA vizsgálata

26



Sebezhetőségi időintervallum a kékkel jelölt kerethez

# ALOHA vizsgálata

Tudjuk, hogy:

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

Azaz most  $t = 2T_f$  és  $k = 0$  ( $t$  legyen a seb. idő,  $k=0$ , hogy ne érkezzen új keret a kék küldése során)

$$P_0(2T_f) = \frac{(\lambda \cdot 2T_f)^0 e^{-\lambda 2T_f}}{0!} = e^{-2G}$$

because  $\lambda = \frac{G}{T_f}$ . Thus,  $S = G \cdot e^{-2G}$

# ALOHA vizsgálata

28

- $S(G) = Ge^{-2G}$  függvényt  $G$  szerint deriválva és az eredményt nullának tekintve az egyenlet megoldásával megkapjuk a maximális sikeres átvitelhez tartozó  $G$  értéket:

$$G = 0.5,$$

melyre  $S(G) = 1/2e = 0.18$ . Azaz a maximális throughput **csak 18%-a** a teljes kapacitásnak!!!

# ALOHA vs TDMA

29

□ A TDMA

□ A v

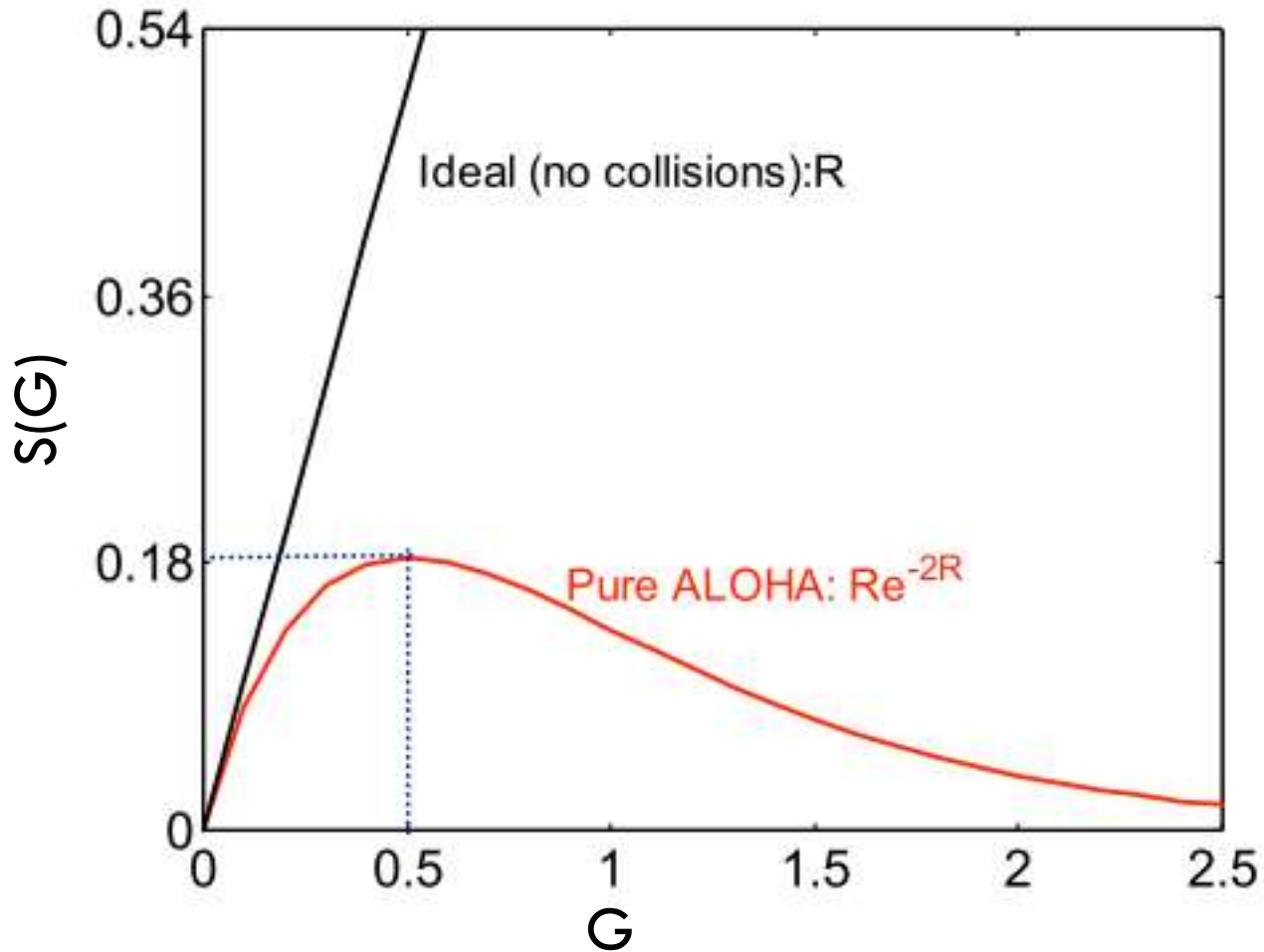
□ Az A

□ Sol

□ De

Sender

Sender

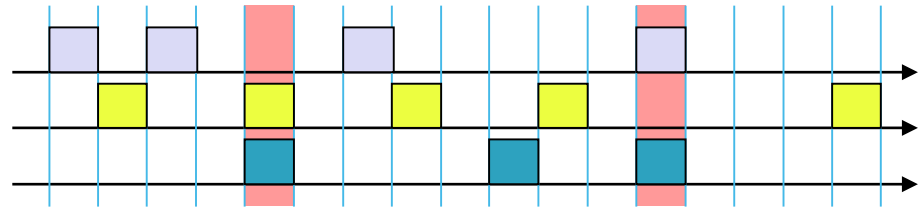


□ Maximum system capacity is 18% of the channel capacity

# Réselt ALOHA

30

- A csatornát azonos időrésekre bontjuk, melyek hossza pont egy keret átviteléhez szükséges idő.
- Átvitel csak az időrések határán lehetséges



- Algoritmus:
  - ▣ Amikor egy új A keret küldésre kész:
    - Az A keret kiküldésre kerül a (következő) időrés-határon

# A réselt ALOHA vizsgálata

□ A sebezhetőségi idő a felére csökken!!!

□ Tudjuk, hogy:

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

Ez esetben  $t = T_f$  és továbbra is  $k = 0$ , amiből kapjuk, hogy:

$$P_0(T_f) = \frac{(\lambda \cdot T_f)^0 e^{-\lambda T_f}}{0!} = e^{-G}$$

because  $\lambda = \frac{G}{T_f}$ . Thus,  $S = G \cdot e^{-G}$

# Réselt ALOHA

32

□ Protokoll

□ Ugrás

■ R

□ Cs

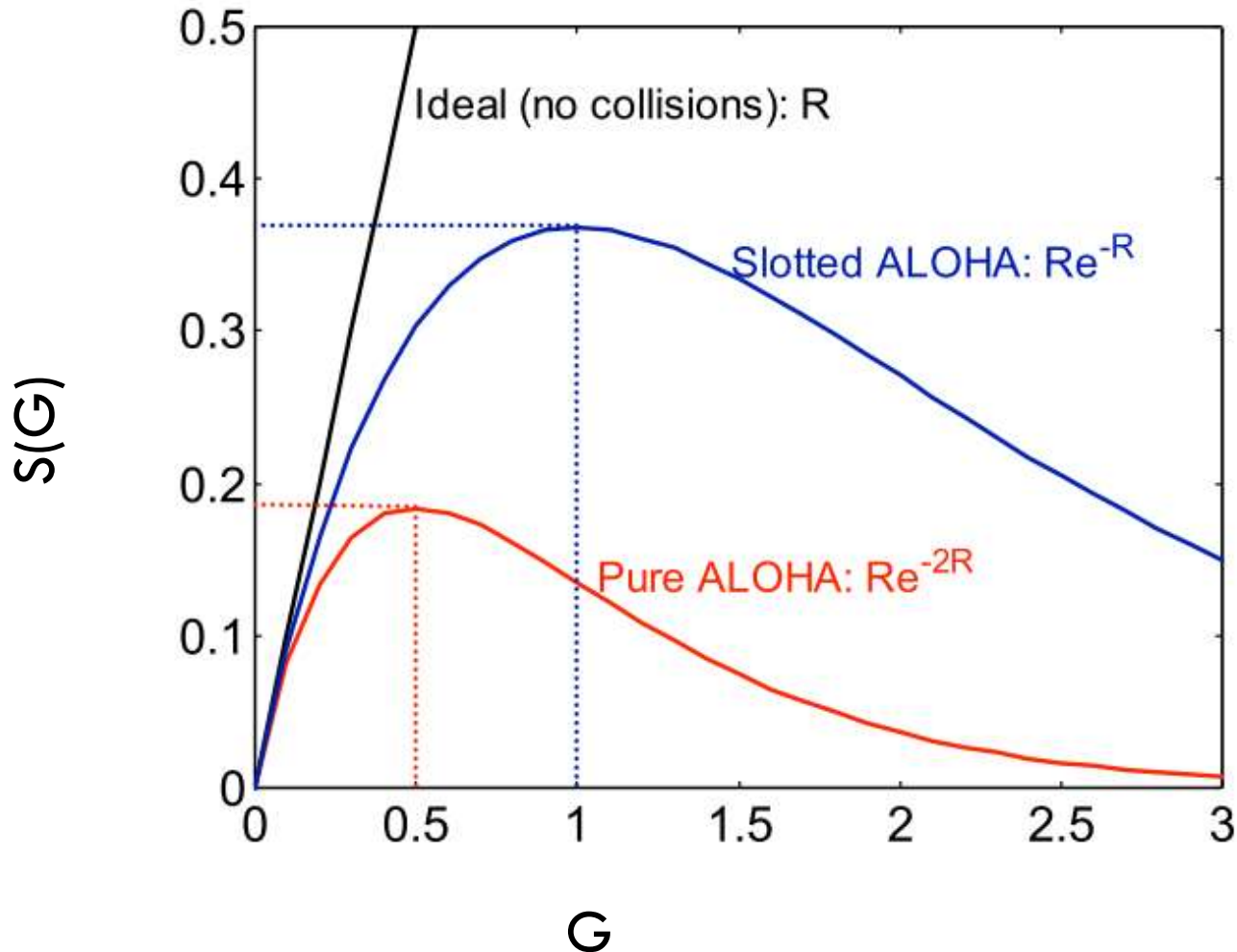
□ Azaz

nem

□ 37

□ Az

ke



általán

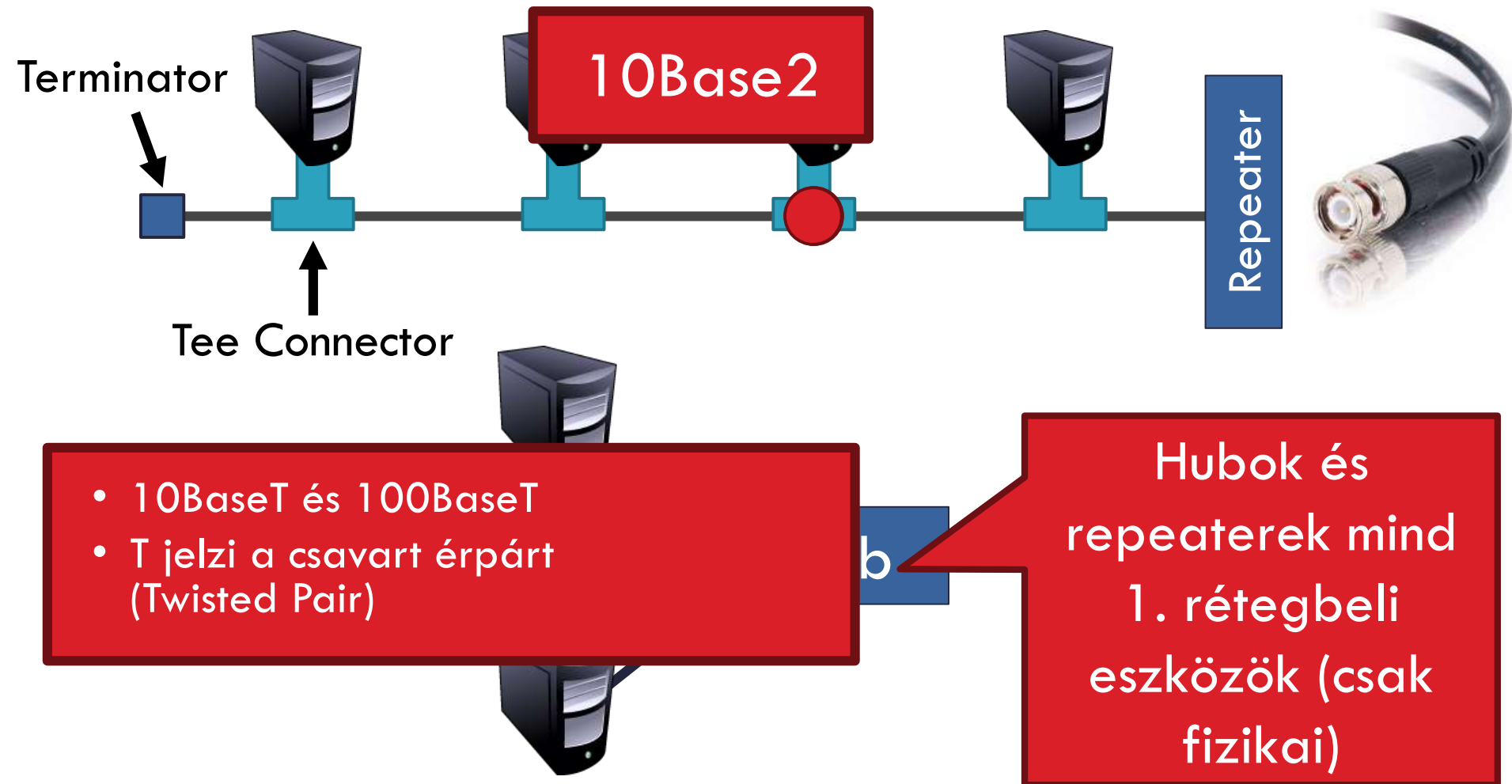
órával



# Adatszóró (Broadcast) Ethernet

33

- Eredetileg az Ethernet egy adatszóró technológia volt



# Vivőjel érzékelés

## Carrier Sense Multiple Access (CSMA)

- További feltételezés

- ▣ Minden állomás képes beleszállgatni a csatornába és így el tudja dönteni, hogy azt más állomás használja-e átvitelre

# 1-perzisztens CSMA protokoll

35

- Vivőjel érzékelés van, azaz minden állomás belehallgathat a csatornába.
- Folytonos időmodellt használ a protokoll

## Algoritmus

- Keret leadása előtt belehallgat a csatornába:
  - a) Ha foglalt, akkor addig vár, amíg fel nem szabadul. Szabad csatorna esetén azonnal küld. (*perzisztens*)
  - b) Ha szabad, akkor küld.
- Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újratekzi a keret leadását.

## Tulajdonságok

- A terjedési késleltetés nagymértékben befolyásolhatja a teljesítményét.
- Jobb teljesítményt mutat, mint az ALOHA protokollok.

# Nem-perzisztens CSMA protokoll

36

- Vivőjel érzékelés van, azaz minden állomás belehallgathat a csatornába.
- Folytonos időmodellt használ a protokoll
- Mohóság kerülése

## Algoritmus

- Keret leadása előtt belehallgat a csatornába:
  - a) Ha foglalt, akkor véletlen ideig vár (nem figyeli a forgalmat), majd kezdi előről a küldési algoritmust. (*nem-perzisztens*)
  - b) Ha szabad, akkor küld.
- Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újratekdi a keret leadását.

## Tulajdonságok

- Jobb teljesítményt mutat, mint az 1-perzisztens CSMA protokoll. (*intuitív*)

# p-perzisztens CSMA protokoll

37

- Vivőjel érzékelés van, azaz minden állomás belehallgathat a csatornába.
- Diszkrét időmodellt használ a protokoll

## Algoritmus

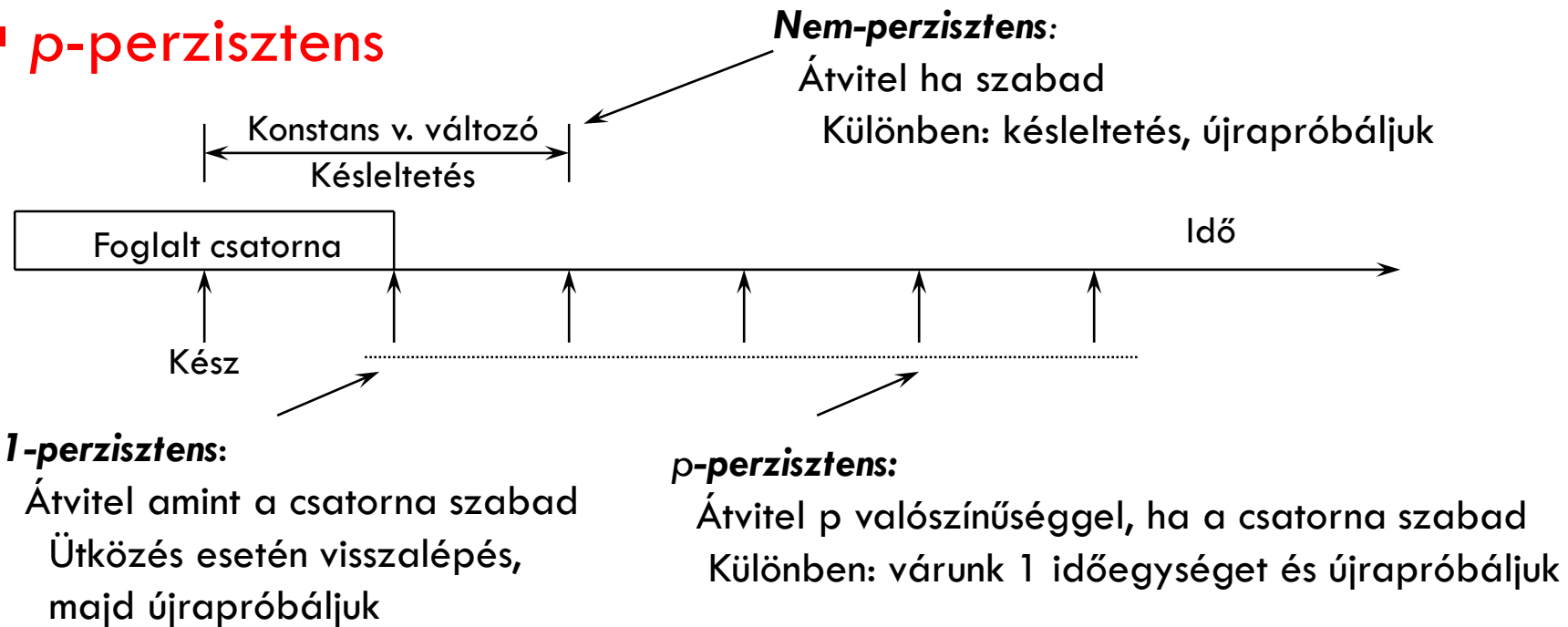
- Adás kész állapotban az állomás belehallgat a csatornába:
  - a) Ha foglalt, akkor vár a következő időrésig, majd megismétli az algoritmust.
  - b) Ha szabad, akkor  $p$  valószínűséggel küld, illetve  $1-p$  valószínűséggel visszalép a szándékától a következő időrésig. Várakozás esetén a következő időrésben megismétli az algoritmust. Ez addig folytatódik, amíg el nem küldi a keretet, vagy amíg egy másik állomás el nem kezd küldeni, mert ilyenkor úgy viselkedik, mintha ütközés történt volna.
- Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újakezdi a keret leadását.

# CSMA áttekintés

38

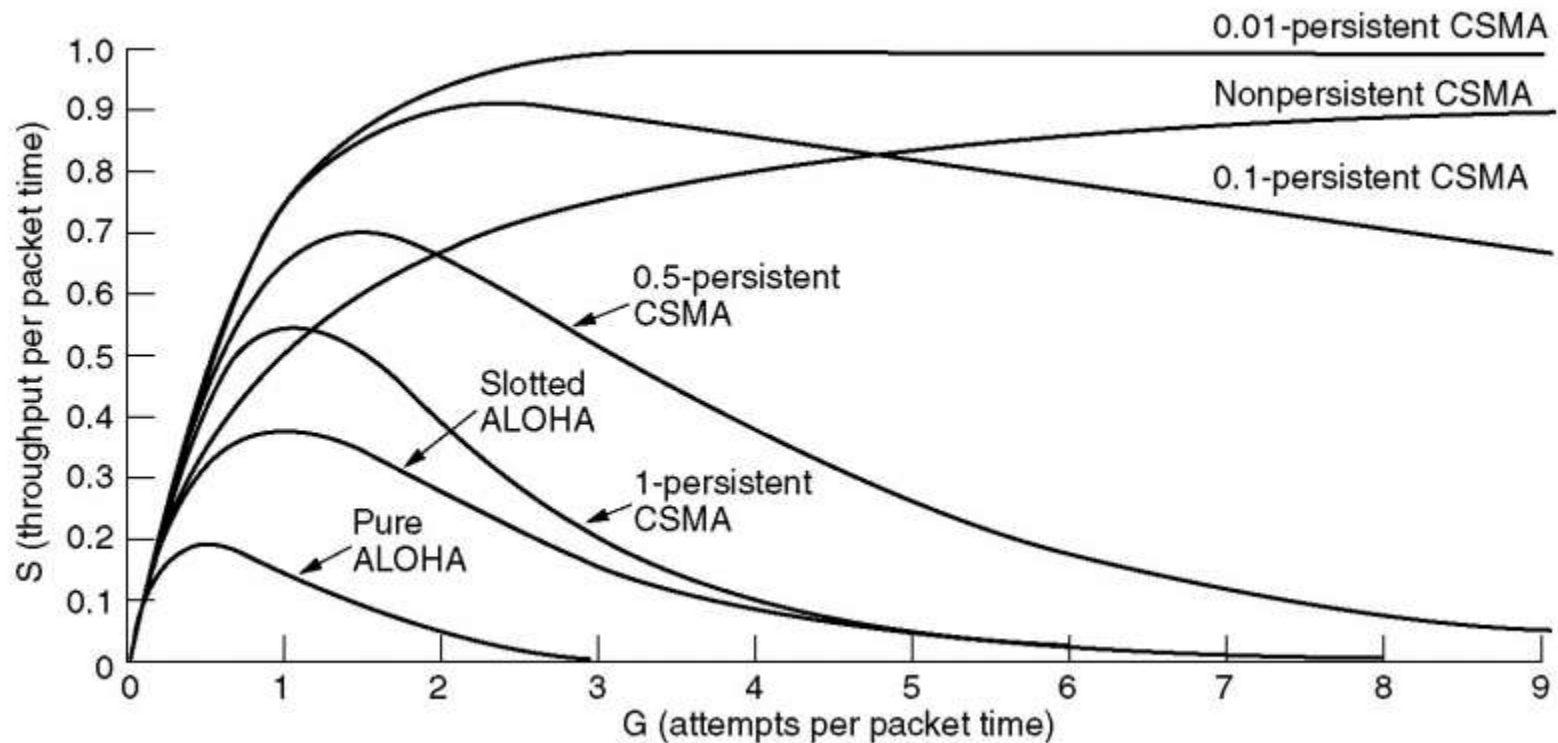
- Nem-perzisztens
- 1-perzisztens
- $p$ -perzisztens

## CSMA perzisztencia



# CSMA és ALOHA protokollok összehasonlítása

39



# CSMA/CD - CSMA ütközés detektálással (CD = Collision Detection)

- Ütközés érzékelés esetén meg lehessen szakítani az adást. („Collision Detection”)
  - ▣ Minden állomás küldés közben megfigyeli a csatornát,
  - ▣ ha ütközést tapasztal, akkor megszakítja az adást, és véletlen ideig várakozik, majd újra elkezdi leadni a keretét.
  
- Mikor lehet egy állomás biztos abban, hogy megszerezte magának a csatornát?
  - ▣ Az ütközés detektálás minimális ideje az az idő, ami egy jelnek a két legtávolabbi állomás közötti átviteléhez szükséges.



# CSMA/CD

- Egy állomás megszerezte a csatornát, ha minden más állomás érzékeli az átvitelét.
- Az **ütközés detektálás működéséhez** szükséges a keretek hosszára egy alsó korlátot adnunk
- Ethernet a CSMA/CD-t használja

# CSMA/CD

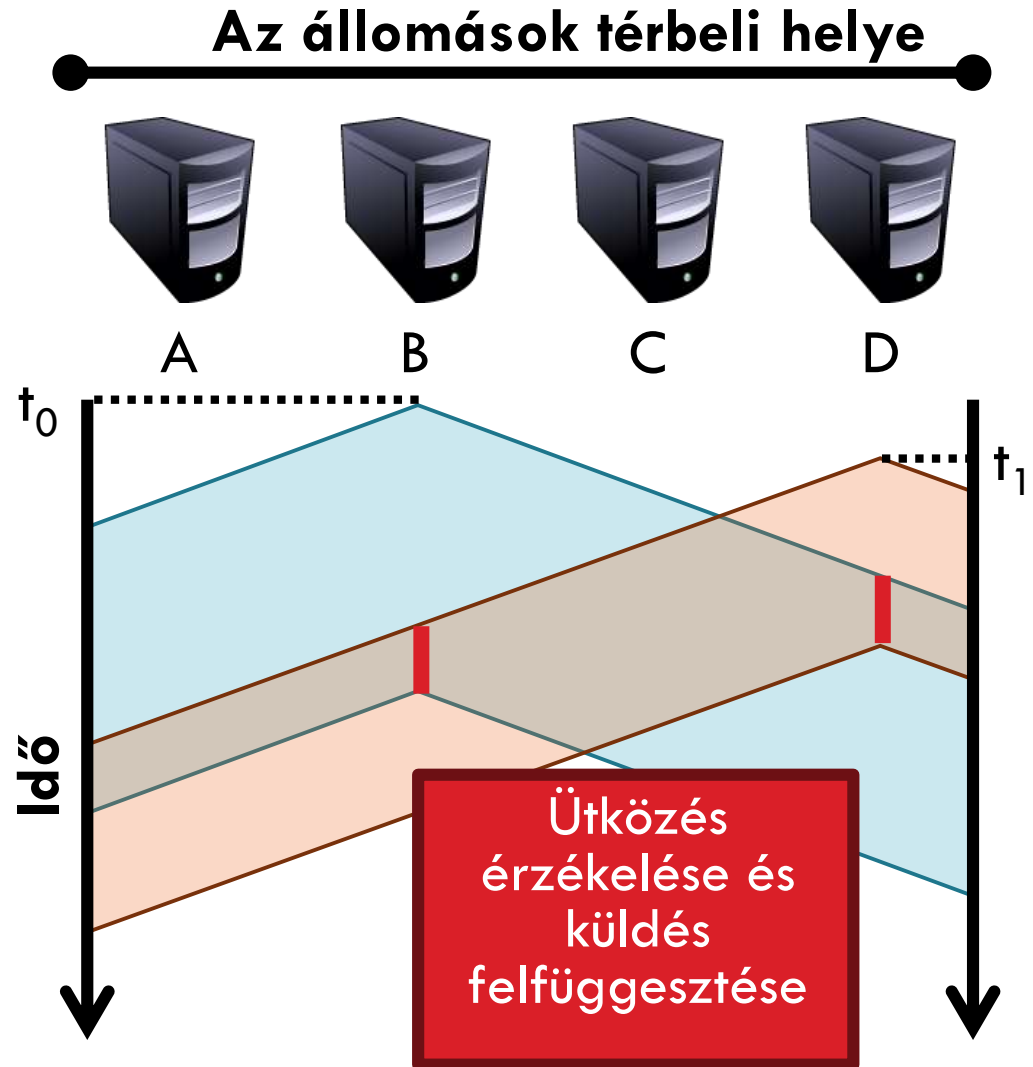
42

- ❑ Carrier sense multiple access with collision detection
- ❑ Alapvetés: a közeg lehetőséget ad a csatornába hallgatásra
- ❑ Algoritmus
  1. Használjuk valamely CSMA variánst
  2. A keret kiküldése után, figyeljük a közeget, hogy történik-e ütközés
  3. Ha nem volt ütközés, akkor a keretet leszállítottuk
  4. Ha ütközés történt, akkor azonnal megszakítjuk a küldést
    - Miért is folytatnánk hisz a keret már sérült...
  5. Alkalmazzuk az bináris exponenciális hátralék módszert az újraküldés során (binary exponential backoff)

# CSMA/CD Ütközések

43

- ❑ Ütközések történhetnek
- ❑ Az ütközéseket gyorsan észleljük és felfüggesztjük az átvitelt
- ❑ Mi a szerepe a távolságnak, propagációs időnek és a keret méretének?



# Binary Exponential Backoff –

## Bináris exponenciális hátralék

44

- Ütközés érzékelésekor a küldő egy ún. „jam” jelet küld
  - ▣ Minden állomás tudomást szerezzen az ütközésről
  
- Binary exponential backoff működése:
  - ▣ Válasszunk egy  $k \in [0, 2^n - 1]$  egyenletes eloszlás szerint, ahol  $n =$  az ütközések száma
  - ▣ Várjunk  $k$  időegységet (keretidőt) az újraküldésig
  - ▣  $n$  felső határa 10, 16 sikertelen próbálkozás után pedig eldobjuk a keretet
  
- A hátralék idő versengési résekre van osztva

# Binary Exponential Backoff

45

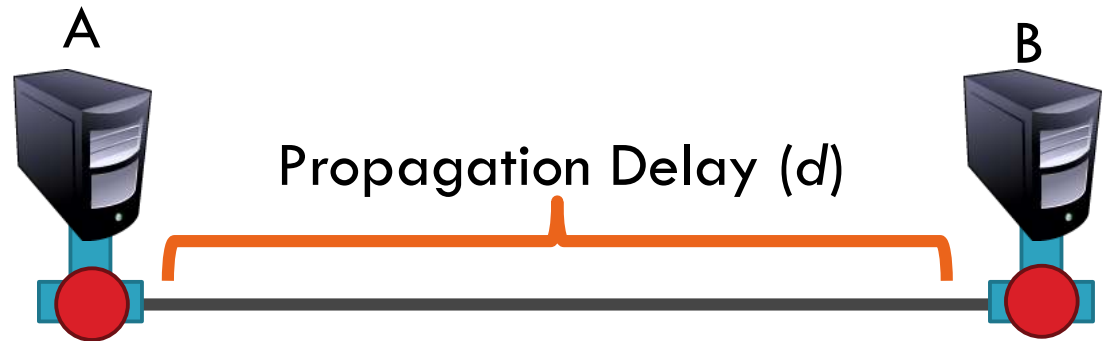
Tekintsünk két állomást, melyek üzenetei ütköztek

- Első ütközés után: válasszunk egyet a két időrés közül
  - ▣ A siker esélye az első ütközés után: 50%
  - ▣ Átlagos várakozási idő: 1,5 időrés
- Második ütközés után: válasszunk egyet a négy rés közül
  - ▣ Sikeres átvitel esélye ekkor: 75%
  - ▣ Átlagos várakozási idő: 2,5 rés
- Általában az  $m$ . ütközés után:
  - ▣ A sikeres átvitel esélye:  $1 - 2^{-m}$
  - ▣ Average delay (in slots):  $0,5 + 2^{(m-1)}$

# Minimális keretméret

46

- Miért 64 bájt a minimális keretméret?
  - ▣ Az állomásoknak elég időre van szüksége az ütközés detektálásához
- Mi a kapcsolat a keretméret és a kábelhossz között?
  1.  $t$  időpont: Az A állomás megkezdte az átvitelt
  2.  $t + d$  időpont: A B állomás is megkezdte az átvitelt
  3.  $t + 2*d$  időpont: A érzékeli az ütközést



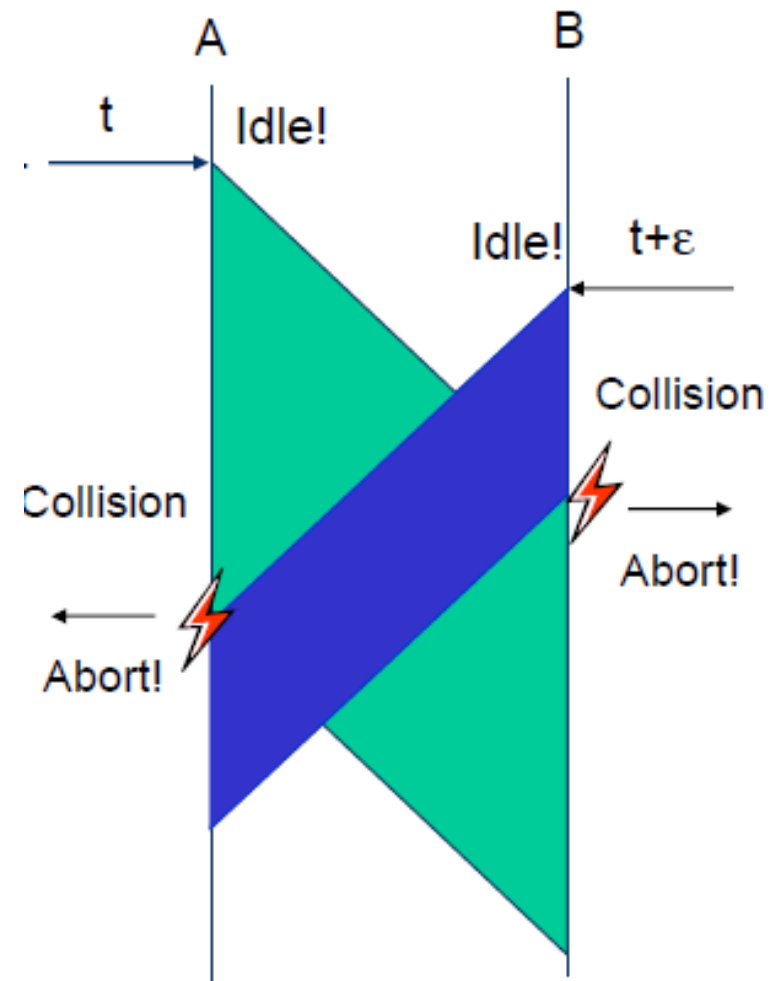
Alapötlet: Az A állomásnak  $2*d$  ideig kell küldenie!

# CSMA/CD

- CSMA/CD három állapota: versengés, átvitel és szabad.
- Ahhoz, hogy minden ütközést észleljünk szükséges:

$$T_f \geq 2T_{pg}$$

- ahol  $T_f$  egy keret elküldéséhez szükséges idő
- és  $T_{pg}$  a propagációs késés A és B állomások között



# Minimális keretméret

48

- Az A küldésének  $2 \cdot d$  ideig kell tartania

- $\text{Min\_keret} = \text{ráta (b/s)} \cdot 2 \cdot d \text{ (s)}$

- ...

- 10 Mbps Ethernet

- Pr
- A keretméret és a kábelhossz változik (m/s)

- Aza a gyorsabb szabványokkal...

- $\text{Min\_keret} = \text{ráta} \cdot 2 \cdot \text{távolság (m)} / \text{fényseb. (m/s)}$

- Azaz a kábel össza ....

- $\text{Távolság} = \text{min\_keret} \cdot \text{fénysebesség} / (2 \cdot \text{ráta})$

$$(64 \text{B} \cdot 8) \cdot (2 \cdot 10^8 \text{mps}) / (2 \cdot 10^7 \text{bps}) = 5120 \text{ méter}$$



# Minimális keretméret

49

- Az A küldésének  $2 \cdot d$  ideig kell tartania
  - ▣  $\text{Min\_keret} = \text{ráta (b/s)} * 2 * d \text{ (s)}$ 
    - ... de mi az a  $d$ ? propagációs késés, melyet a fénysebesség ismeretében ki tudunk számolni
    - $\text{Propagációs késés (d)} = \text{távolság (m)} / \text{fénysebesség (m/s)}$
  - ▣ Azaz:
  - ▣  $\text{Min\_keret} = \text{ráta (b/s)} * 2 * \text{távolság (m)} / \text{fényseb. (m/s)}$
- Azaz a kábel összhossza ....
  - ▣  $\text{Távolság} = \text{min\_keret} * \text{fénysebesség} / (2 * \text{ráta})$

$$(64\text{B} * 8) * (2 * 10^8 \text{mps}) / (2 * 10^7 \text{bps}) = 5120 \text{ méter}$$

# Kábelhossz példa

50

$$\text{min\_keret} * \text{fénysebesség} / (2 * \text{ráta}) = \text{max\_kábelhossz}$$
$$(64\text{B} * 8) * (2 * 10^8 \text{mps}) / (2 * 10 \text{Mbps}) = 5120 \text{ méter}$$

- Mi a maximális kábelhossz, ha a minimális keretméret 1024 bájtra változik?
  - ▣ 81,9 kilométer
- Mi a maximális kábelhossz, ha a ráta 1 Gbps-ra változik?
  - ▣ 51 méter
- Mi történik, ha mindkettő változik egyszerre?
  - ▣ 819 méter

# Maximális keretméret

51

- ❑ Maximum Transmission Unit (MTU): 1500 bájt
- ❑ Pro:
  - ▣ Hosszú csomagokban levő biz hibák jelentős javítási költséget okozhatnak (pl. túl sok adatot kell újraküldeni)
- ❑ Kontra:
  - ▣ Több bájtot vesztegetünk el a fejlécekben
  - ▣ Összességében nagyobb csomag feldolgozási idő
- ❑ Adatközpontokban Jumbo keretek
  - ▣ 9000 bájtos keretek

# Ütközésmentes protokollok

52

## MOTIVÁCIÓ

- az ütközések hátrányosan hatnak a rendszer teljesítményére
  - ▣ hosszú kábel, rövid keret
- a CSMA/CD nem mindenhol alkalmazható

## FELTÉTELEZÉSEK

- N állomás van.
- Az állomások 0-ától N-ig egyértelműen sorszámozva vannak.
- Réselt időmodellt feltételezünk.

# Alapvető bittérkép protokoll

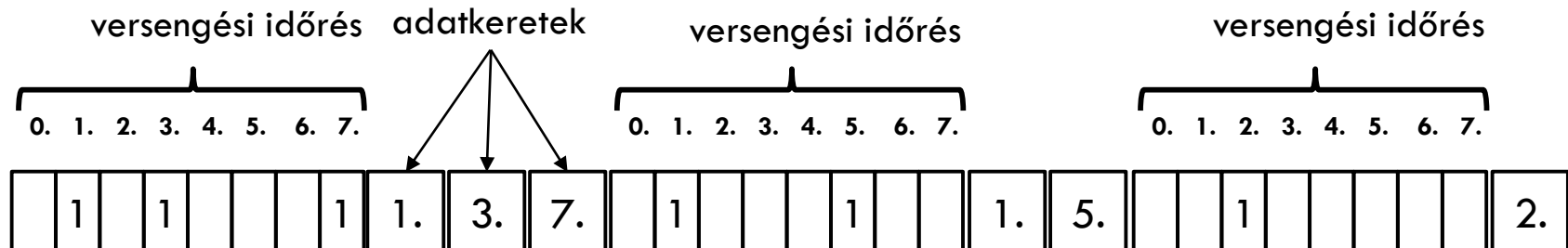
## - Egy helyfoglalásos megoldás

53

- alapvető bittérkép eljárás

### MŰKÖDÉS

- Az ütköztetési periódus  $N$  időrés
- Ha az  $i$ -edik állomás küldeni szeretne, akkor a  $i$ -edik versengési időrésben egy 1-es bit elküldésével jelezheti. (adatszórás)
- A versengési időszak végére minden állomás ismeri a küldőket. A küldés a sorszámok szerinti sorrendben történik meg.



# Bináris visszaszámlálás protokoll 1 / 2

54

- alapvető bittérkép eljárás hátrány, hogy az állomások számának növekedésével a versengési periódus hossza is nő

## MŰKÖDÉS

- Minden állomás azonos hosszú bináris azonosítóval rendelkezik.
- A forgalmazni kívánó állomás elkezd a bináris címét bitenként elküldeni a legnagyobb helyi értékű bittel kezdve. Az azonos pozíciójú bitek logikai VAGY kapcsolatba lépnek ütközés esetén. Ha az állomás nullát küld, de egyet hall vissza, akkor feladja a küldési szándékát, mert van nála nagyobb azonosítóval rendelkező küldő.

|                |   |   |   |   |
|----------------|---|---|---|---|
| A HOSZT (0011) | 0 | – | – | – |
| B HOSZT (0110) | 0 | – | – | – |
|                | 1 | 0 | 1 | 0 |
| C HOSZT (1010) | 1 | 0 | 1 | 1 |
| D HOSZT (1011) | 1 | 0 | 1 | 1 |

D kerete

# Bináris visszaszámlálás protokoll 2/2

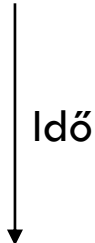
55

- **Következmény:** a magasabb címmel rendelkező állomásoknak a prioritásuk is magasabb az alacsonyabb című állomásokénál

## MOK ÉS WARD MÓDOSÍTÁSA

- Virtuális állomás címek használata.
- Minden sikeres átvitel után ciklikusan permutáljuk az állomások címét.

|                 | A   | B   | C   | D   | E   | F   | G   | H   |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Kezdeti állapot | 100 | 010 | 111 | 101 | 001 | 000 | 011 | 110 |



# Korlátozott versenyes protokollok

56

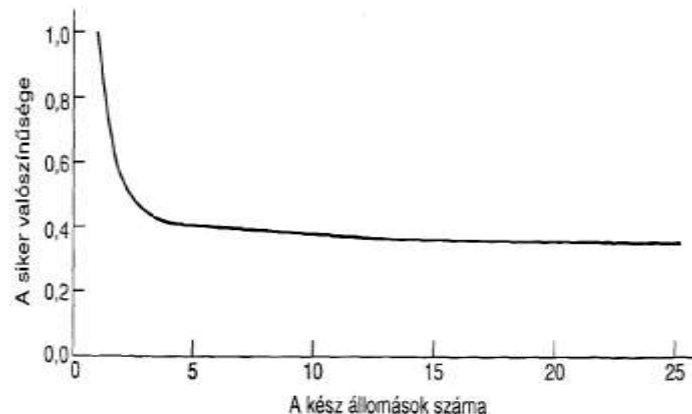
- **Cél:** Ötvözni a versenyhelyzetes és ütközésmentes protokollok jó tulajdonságait.
- **korlátozott versenyes protokoll** – Olyan protokoll, amely kis terhelés esetén versenyhelyzetes technikát használ a kis késleltetés érdekében, illetve nagy terhelés mellett ütközésmentes technikát alkalmaz a csatorna jó kihasználása érdekében.

## SZIMMETRIKUS PROTOKOLLOK

- Adott részben  $k$  állomás verseng, minden állomás  $p$  valószínűséggel adhat. A csatorna megszerzésének valószínűsége:  $kp(1 - p)^{k-1}$ .

$$P(\text{siker optimális } p \text{ mellett}) = \left(\frac{k-1}{k}\right)^{k-1}$$

- Azaz a csatorna megszerzésének esélyeit a versenyhelyzetek számának csökkentésével érhetjük el.



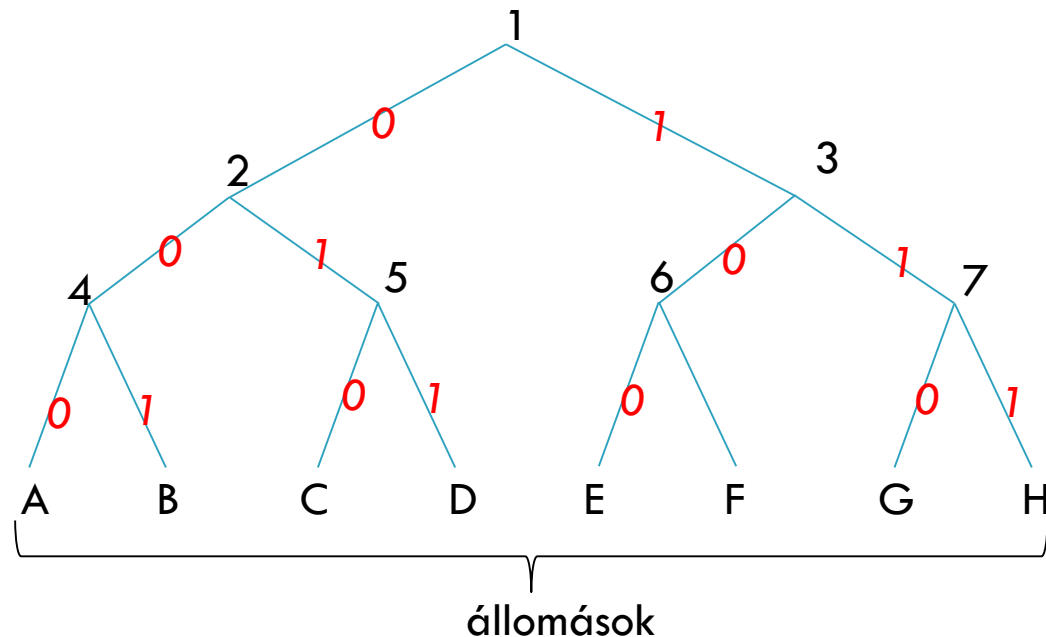


# Adaptív fabejárési protokoll 1/2

57

## Történeti háttér

- 1943 – Dorfman a katonák szifilisz fertőzöttségét vizsgálta.
- 1979 – Capetanakis bináris fa reprezentáció az algoritmus számítógépes változatával.



# Adaptív fabejárási protokoll 2/2

58

## Működés

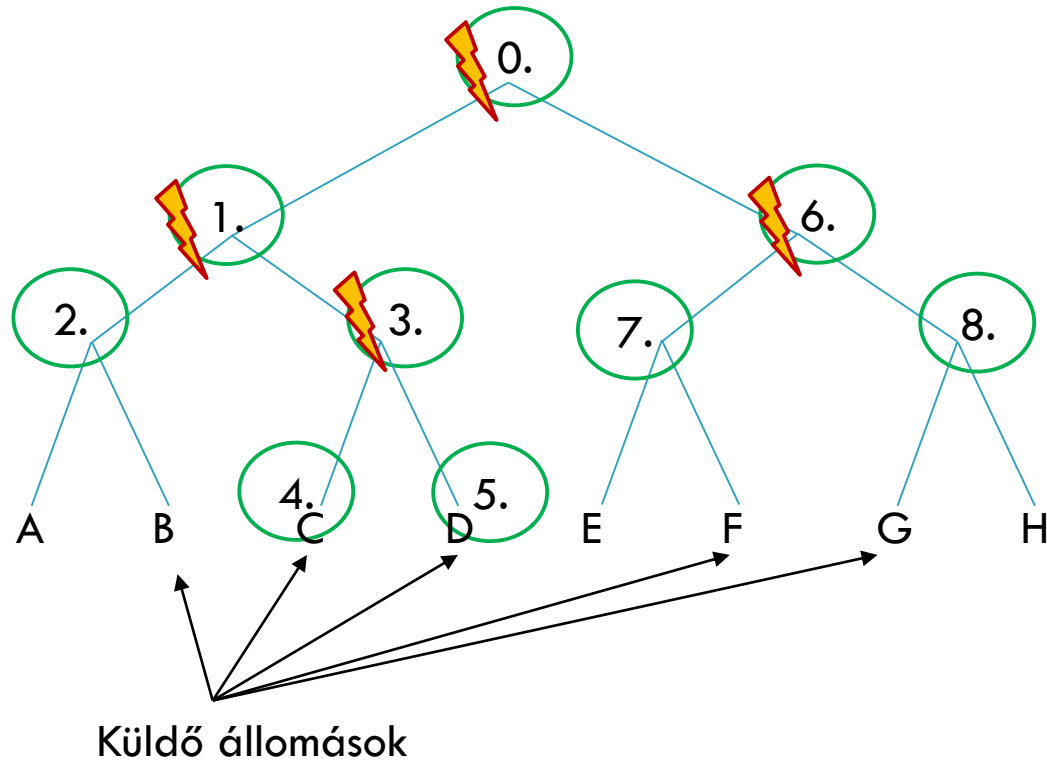
- 0-adik időrésben mindenki küldhet.
  - ▣ Ha ütközés történik, akkor megkezdődik a fa *mélységi bejárása*.
- A rések a fa egyes csomópontjaihoz vannak rendelve.
- Ütközéskor rekurzívan az adott csomópont bal illetve jobb gyerekcsomópontjánál folytatódik a keresés.
- Ha egy bitrés kihasználatlan marad, vagy pontosan egy állomás küld, akkor a szóban forgó csomópont keresése befejeződik.

## Következmény

- Minél nagyobb a terhelés, annál mélyebben érdemes kezdeni a keresést.

# Adaptív fabejárás példa

59



# Az adatkapcsolati réteg „legtetején”...

60

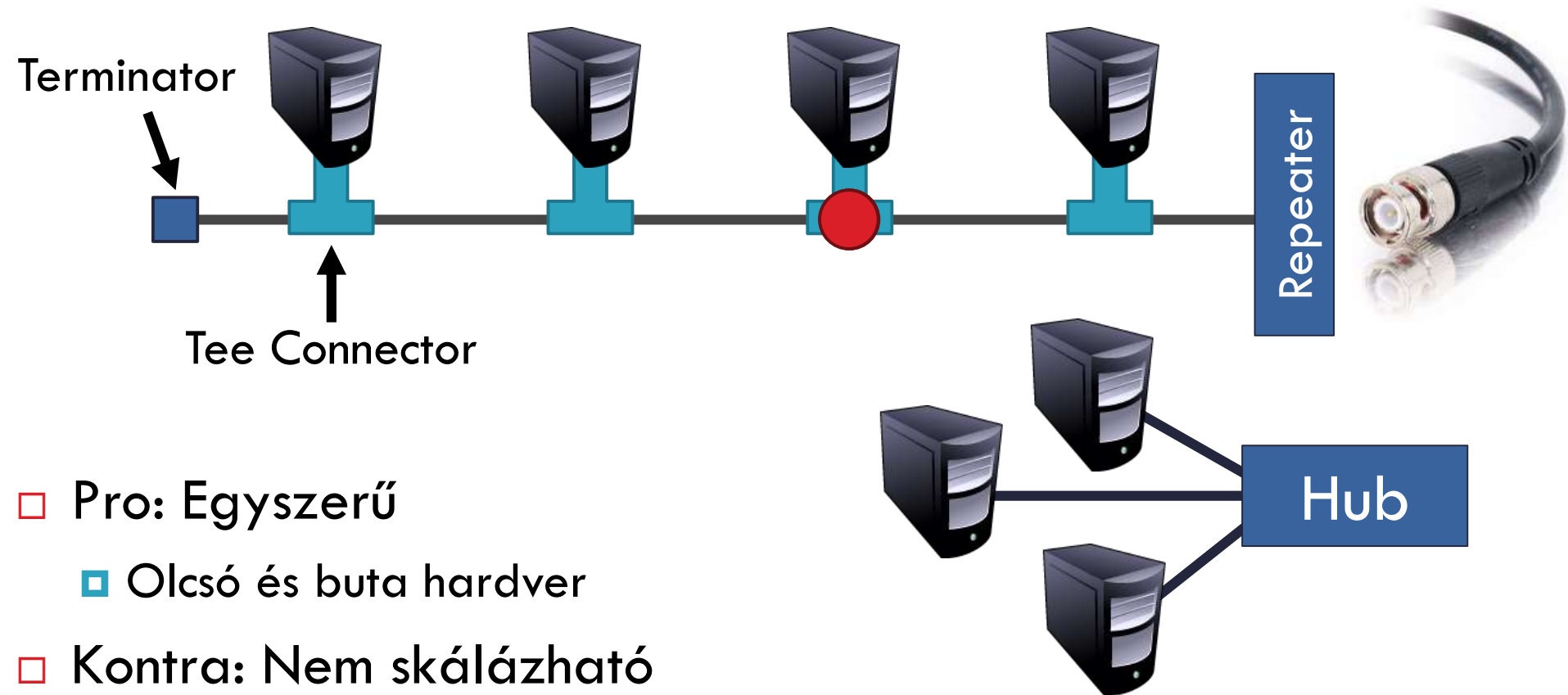


- Bridging, avagy hidak
  - ▣ Hogyan kapcsoljunk össze LAN-okat?
- Funkciók:
  - ▣ Keretek forgalomirányítása a LAN-ok között
- Kihívások:
  - ▣ Plug-and-play, önmagát konfiguráló
  - ▣ Esetleges hurkok feloldása

# Visszatekintés

61

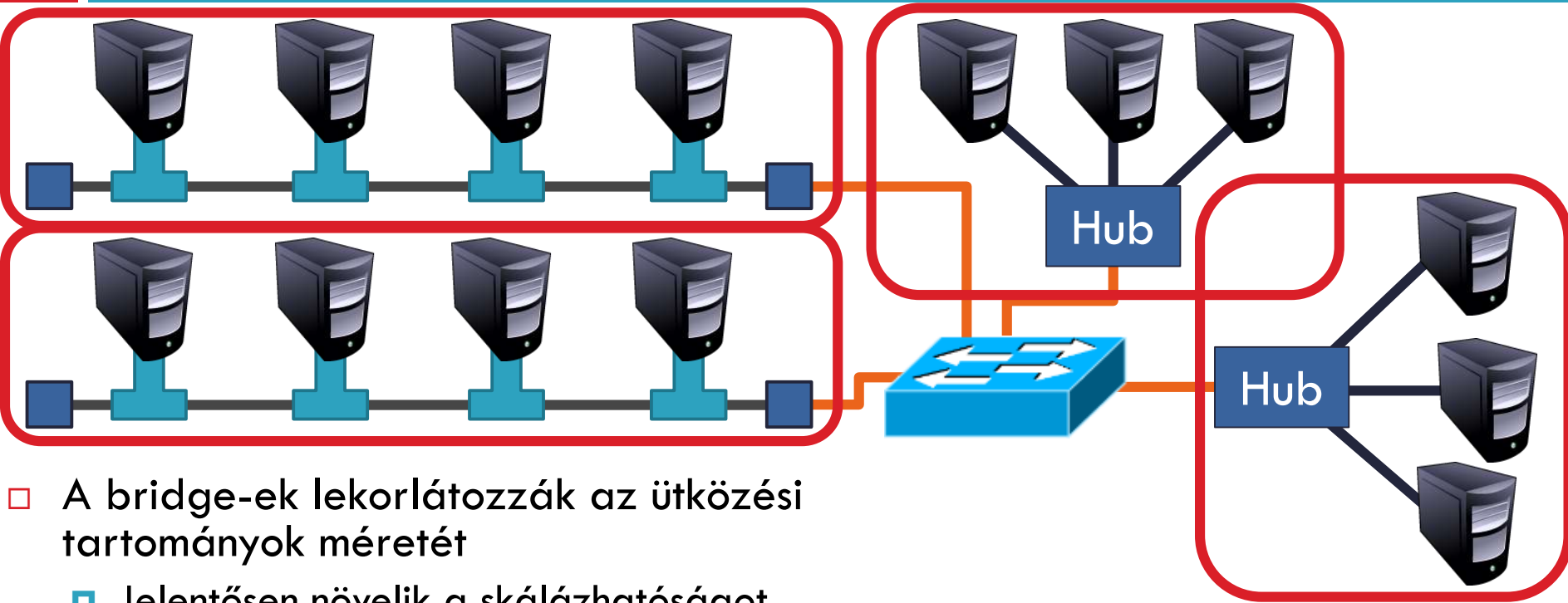
- Az Ethernet eredetileg adatszóró technológia volt



- Pro: Egyszerű
  - ▣ Olcsó és buta hardver
- Kontra: Nem skálázható
  - ▣ Több állomás = több ütközés = káosz

# LAN-ok összekapcsolása

62

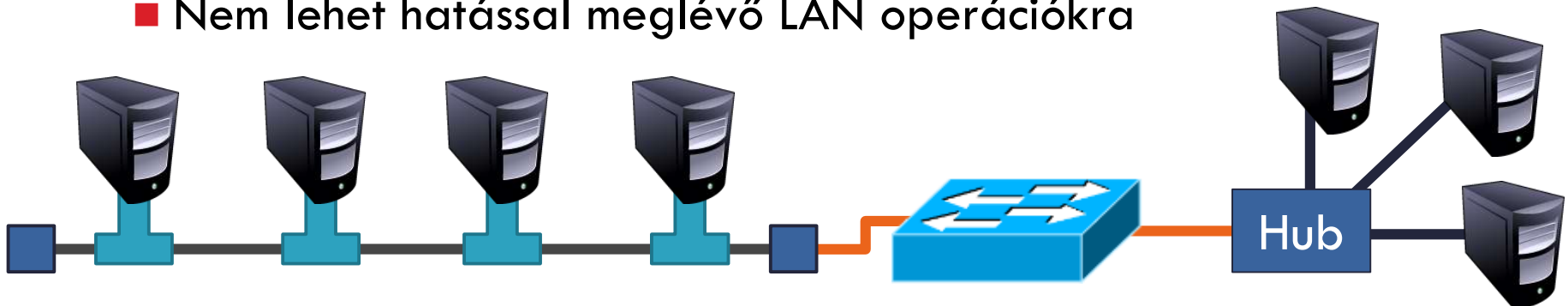


- ❑ A bridge-ek lekorlátozzák az ütközési tartományok méretét
  - ▣ Jelentősen növelik a skálázhatóságot
  - ▣ Kérdés: lehetne-e az egész Internet egy bridge-ekkel összekötött tartomány?
- ❑ Hátrány: a bridge-ek sokkal komplexebb eszközök a hub-oknál
  - ▣ Fizikai réteg VS Adatkapcsolati réteg
  - ▣ Memória pufferek, csomag feldolgozó hardver és routing (útválasztó) táblák szükségesek

# Bridge-ek (magyarul: hidak)

63

- ❑ Az Ethernet switch eredeti formája
- ❑ Több IEEE 802 LAN-t kapcsol össze a 2. rétegben
- ❑ Célok
  - ▣ Ütközési tartományok számának csökkentése
  - ▣ Teljes átlátszóság
    - “Plug-and-play,” önmagát konfiguráló
    - Nem szükségesek hw és sw változtatások a hosztokon/hub-okon
    - Nem lehet hatással meglévő LAN operációkra



# Bridge-ek (magyarul: hidak)

64

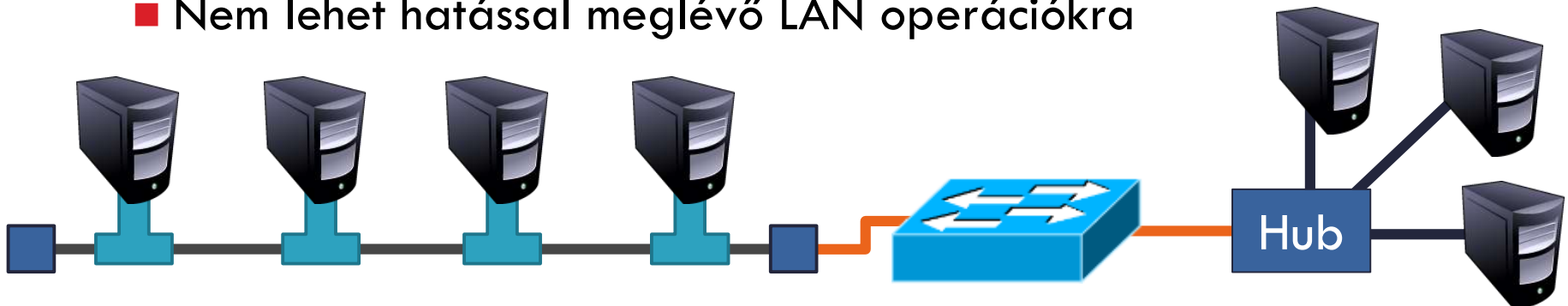
- Az Ethernet switch eredeti formája

□

□

1. Keretek továbbítása
2. (MAC) címek tanulása
3. Feszítőfa (Spanning Tree) Algoritmus (a hurkok kezelésére)

- Nem szükségesek hw és sw változtatások a hosztokon/hub-okon
- Nem lehet hatással meglévő LAN operációkra



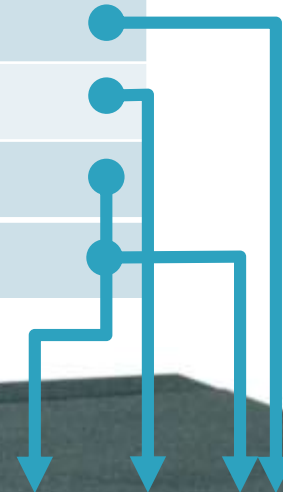


# Keret Továbbító Táblák

65

- Minden bridge karbantart egy továbbító táblát (forwarding table)

| MAC Cím           | Port | Kor    |
|-------------------|------|--------|
| 00:00:00:00:00:AA | 1    | 1 perc |
| 00:00:00:00:00:BB | 2    | 7 perc |
| 00:00:00:00:00:CC | 3    | 2 mp   |
| 00:00:00:00:00:DD | 1    | 3 perc |



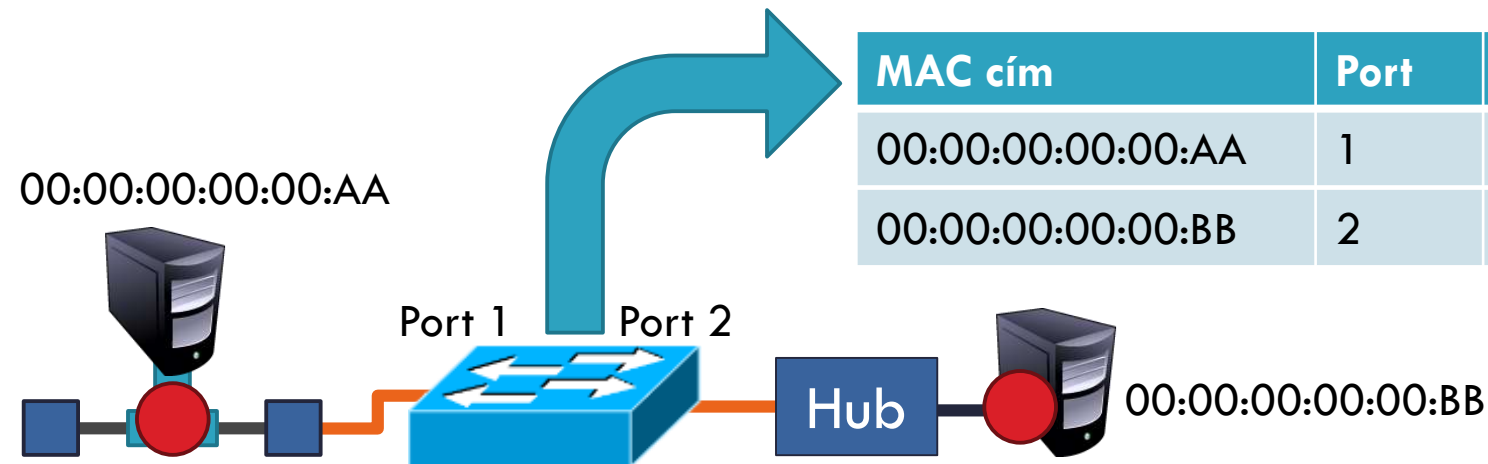
# Címek tanulása

66

- ❑ Kézi beállítás is lehetséges, de...
  - ▣ Időigényes
  - ▣ Potenciális hiba forrás
  - ▣ Nem alkalmazkodik a változásokhoz (új hosztok léphetnek be és régiek hagyhatják el a hálózatot)
- ❑ Ehelyett: tanuljuk meg a címeket
  - ▣ Tekintsük a **forrás címeket** a különböző portokhoz tartozó kereteknek --- képezzünk ebből egy táblázatot

Töröljük a régi bejegyzéseket

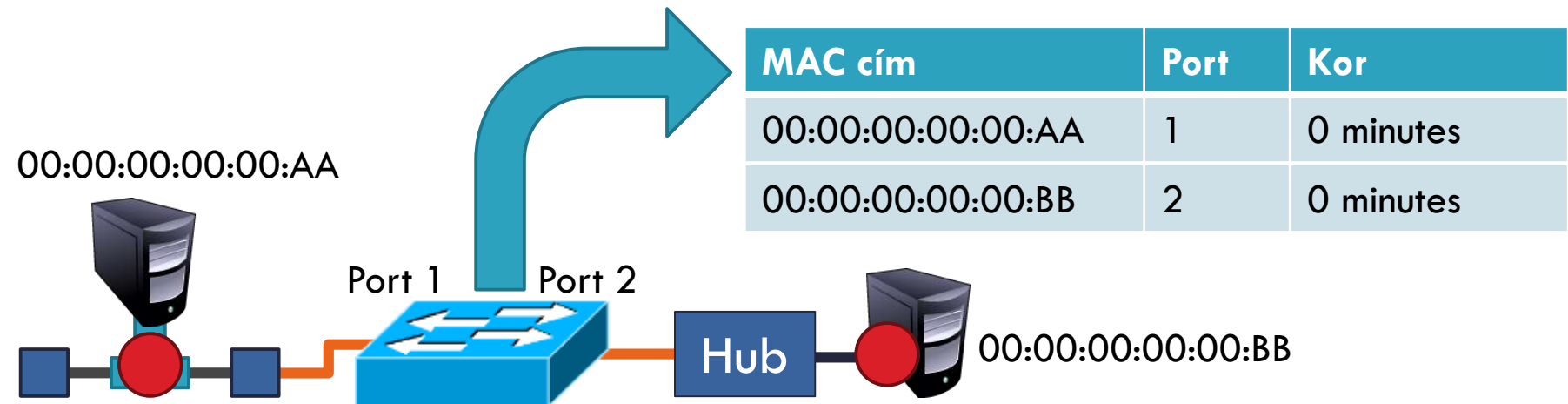
| MAC cím           | Port | Kor       |
|-------------------|------|-----------|
| 00:00:00:00:00:AA | 1    | 0 minutes |
| 00:00:00:00:00:BB | 2    | 0 minutes |



# Címek tanulása

67

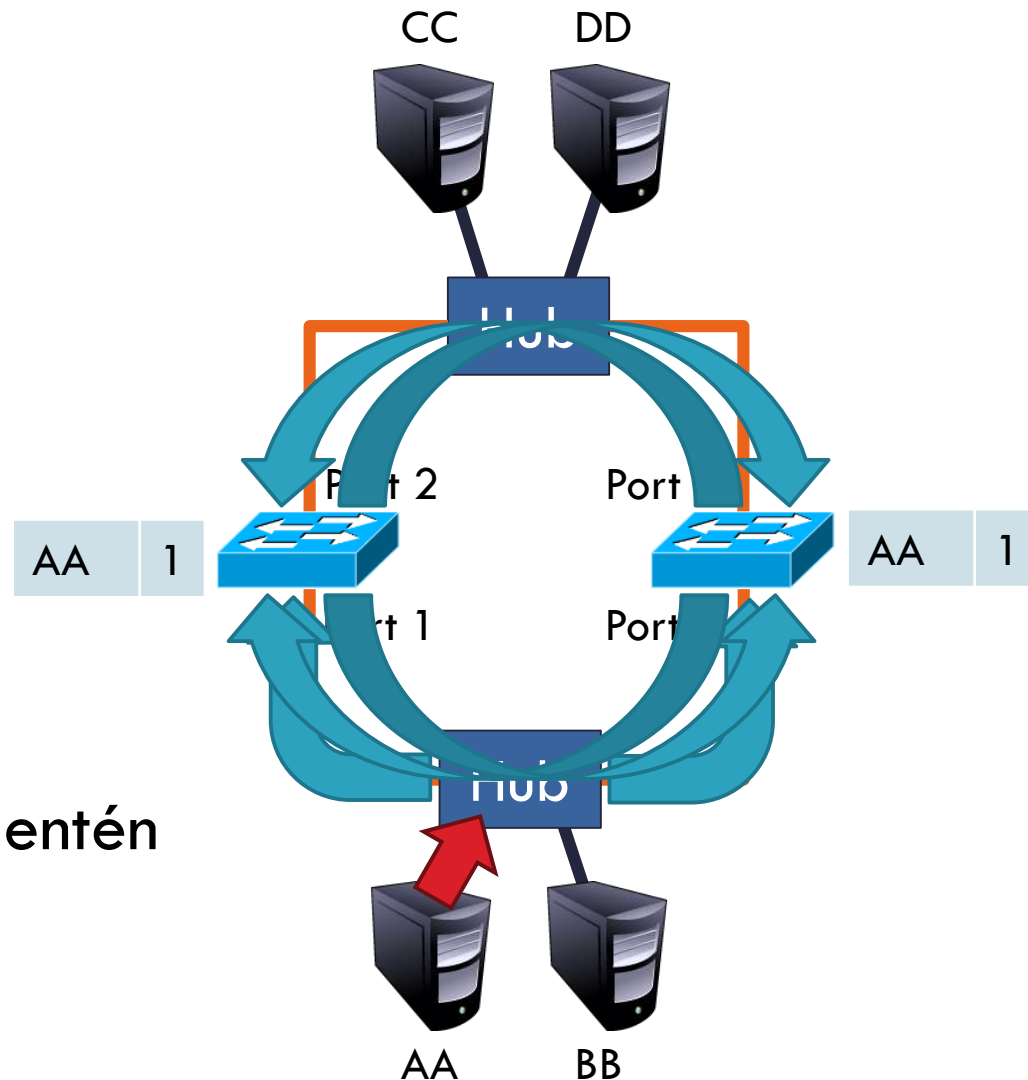
- ❑ Kézi beállítás is lehetséges, de...
  - ▣ Időigényes
  - ▣ Potenciális hiba forrás
  - ▣ Nem alkalmazkodik a változásokhoz (új hosztok léphetnek be és régiek hagyhatják el a hálózatot)
- ❑ Ehelyett: tanuljuk meg a címeket
  - ▣ Tekintsük a **forrás cím** a különböző portokon beérkező kereteknek --- képezzünk ebből egy táblázatot



# Hurkok problémája

68

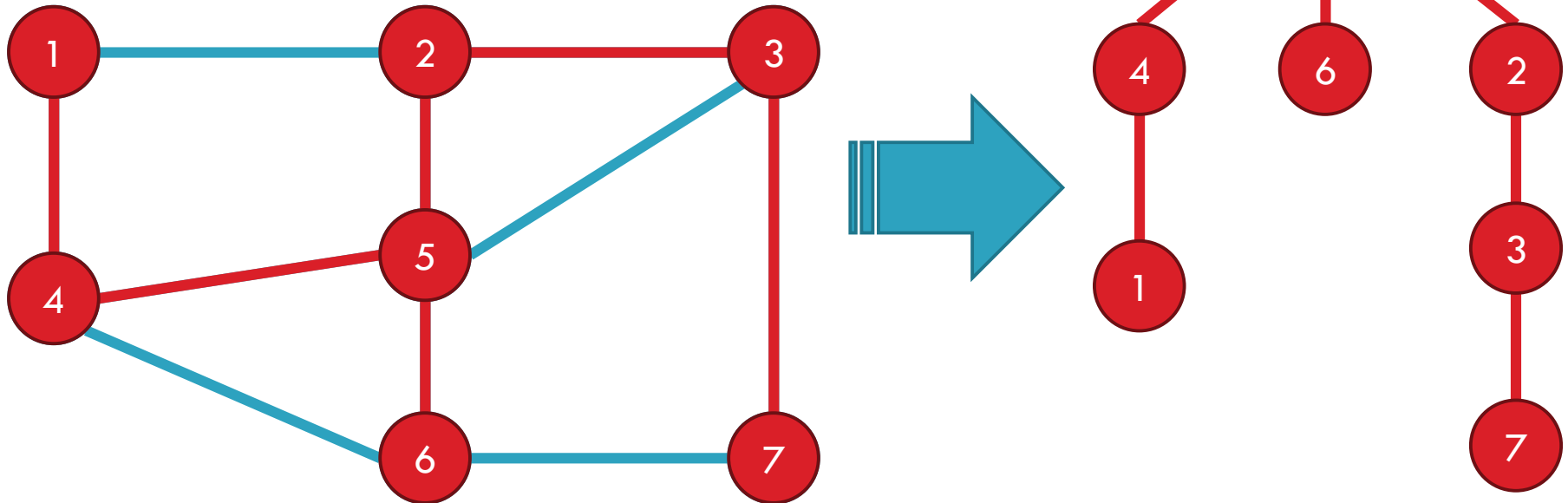
- ❑  $\langle \text{Src}=\text{AA}, \text{Dest}=\text{DD} \rangle$
- ❑ Ez megy a végtelenségig
  - ▣ Hogyan állítható meg?
- ❑ Távolítsuk el a hurkokat a topológiából
  - ▣ A kábelek kihúzása nélkül
- ❑ 802.1 (LAN) definiál egy algoritmust **feszítőfa** építéséhez és karbantartásához, mely mentén lehetséges a keretek továbbítása



# Feszítőfa

69

- Egy gráf éleinek részhalmaza, melyre teljesül:
  - ▣ Lefed minden csomópontot
  - ▣ Nem tartalmaz köröket
- Továbbá a struktúra egy fa-gráf



# A 802.1 feszítőfa algoritmus

70

1. Az egyik bridge-et megválasztjuk a fa gyökerének
  2. Minden bridge megkeresi a legrövidebb utat a gyökérhez
  3. Ezen utak unióját véve megkapjuk a feszítőfát
- 
- A fa építése során a bridge-ek egymás között konfigurációs üzeneteket (Configuration Bridge Protocol Data Units [BPDUs]) cserélnek
    - ▣ A gyökér elem megválasztásához
    - ▣ A legrövidebb utak meghatározásához
    - ▣ A gyökérhez legközelebbi szomszéd (next hop) állomás és a hozzá tartozó port azonosításához
    - ▣ A feszítőfához tartozó portok kiválasztása

# Gyökér meghatározása

71

- Kezdetben minden állomás feltételezi magáról, hogy gyökér
- Bridge-ek minden irányba szétküldik a BPDU üzeneteiket:

Bridge ID

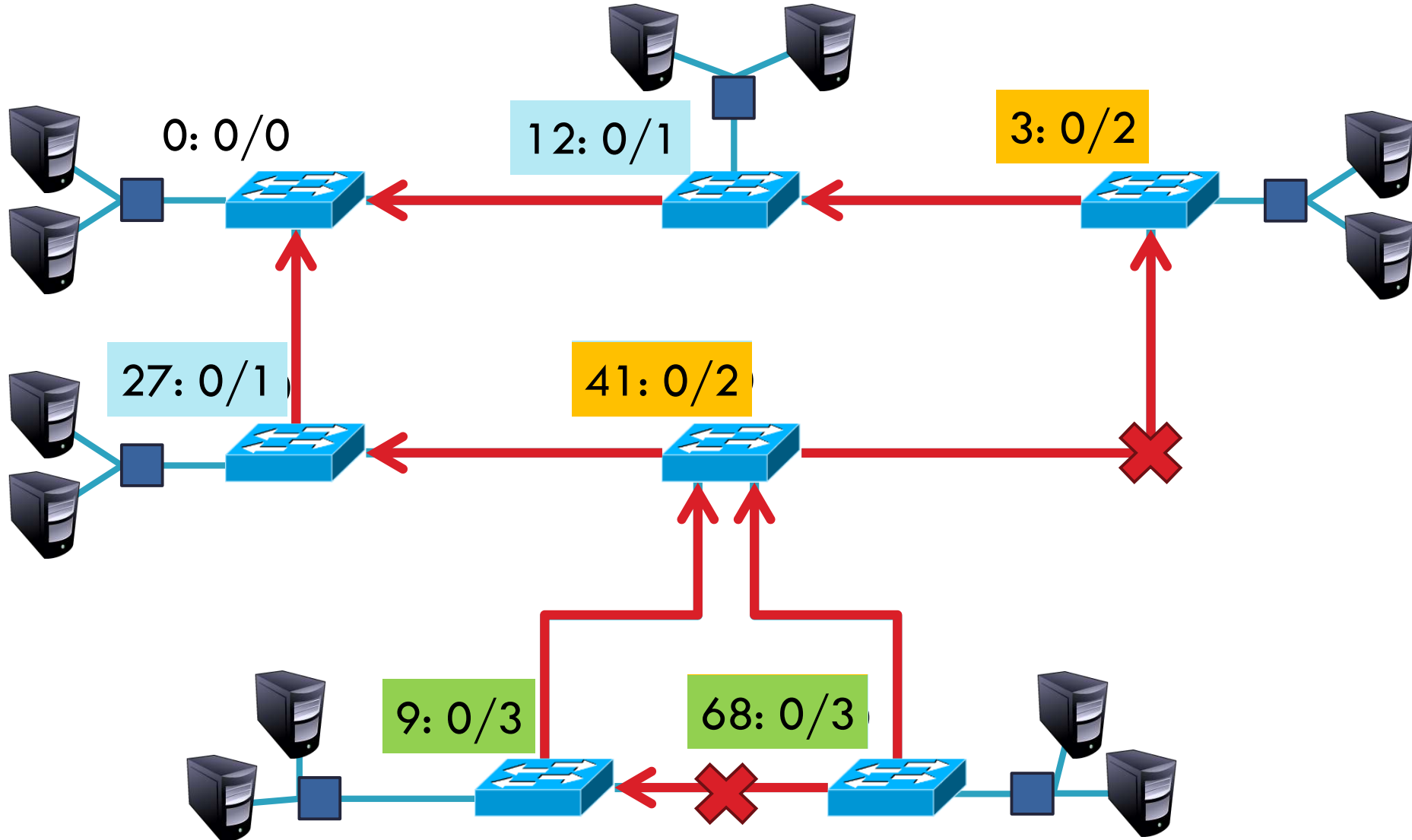
Gyökér ID

Út költség a gyökérhez

- A fogadott BPDU üzenet alapján, minden switch választ:
  - ▣ Egy új gyökér elemet (legkisebb ismert Gyökér ID alapján)
  - ▣ Egy új gyökér portot (melyik interfész megy a gyökér irányába)
  - ▣ Egy új kijelölt bridge-et (a következő állomás a gyökérhez vezető úton)

# Feszítőfa építése

72





# Bridge-ek vs. Switch-ek

## Hidak vs. Kapcsolók

73

- ❑ A bridge-ek lehetővé teszik hogy növeljük a LAN-ok kapacitását
  - ▣ Csökkentik a sikeres átvitelhez szükséges elküldendő csomagok számát
  - ▣ Kezeli a hurkokat
- ❑ A switch-ek a bridge-ek speciális esetei
  - ▣ Minden port egyetlen egy hoszthoz kapcsolódik
    - Lehet egy kliens terminál
    - vagy akár egy másik switch
  - ▣ Full-duplex link-ek
  - ▣ Egyszerűsített hardver: nincs szükség CSMA/CD-re!
  - ▣ Különböző sebességű/rátájú portok is lehetségesek

# Kapcsoljuk össze az Internetet

74

- ❑ Switch-ek képességei:
  - ▣ MAC cím alapú útvonalválasztás a hálózatban
  - ▣ Automatikusan megtanulja az utakat egy új állomáshoz
  - ▣ Feloldja a hurkokat
- ❑ Lehetne a teljes internet egy ily módon összekötött tartomány?

NEM

# Korlátok

75

- ❑ Nem hatékony
  - ▣ Elárasztás ismeretlen állomások megtalálásához
- ❑ Gyenge teljesítmény
  - ▣ A feszítőfa nem foglalkozik a terhelés elosztással
  - ▣ Hot spots
- ❑ Nagyon gyenge skálázhatóság
  - ▣ Minden switch-nek az Internet összes MAC címét ismerni kellene a továbbító táblájában!
- ❑ Az IP fogja ezt a problémát megoldani...

Köszönöm a figyelmet!