

# Számítógépes Hálózatok

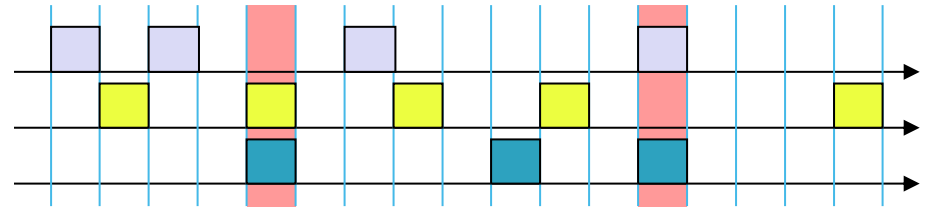
## 5. Előadás: Adatkapcsolati réteg III.

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

# Réselt ALOHA

2

- 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

4

□ Protokoll

□ Ugrás

■ R

□ Cs

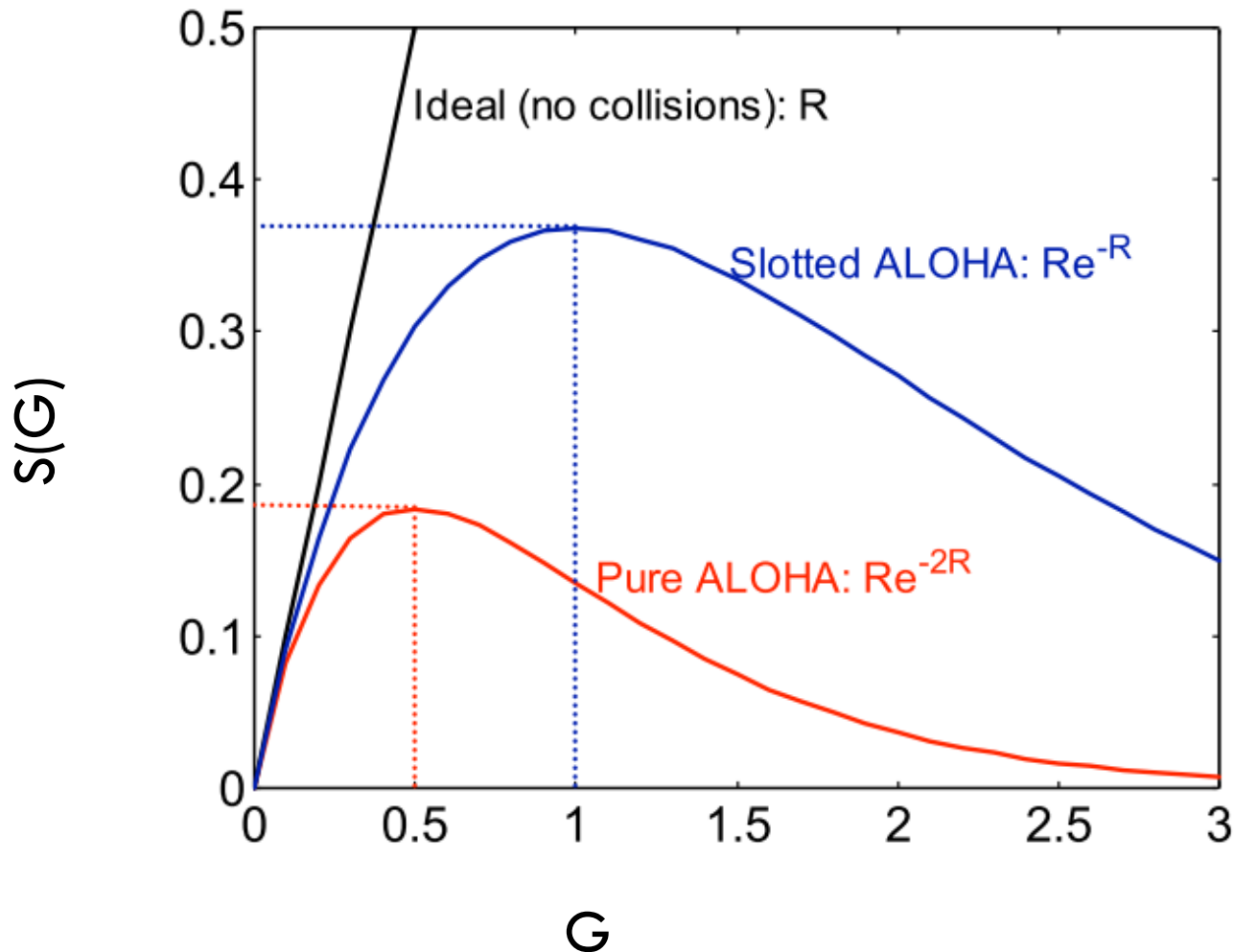
□ Azaz

nem

□ 37

□ Az

ke



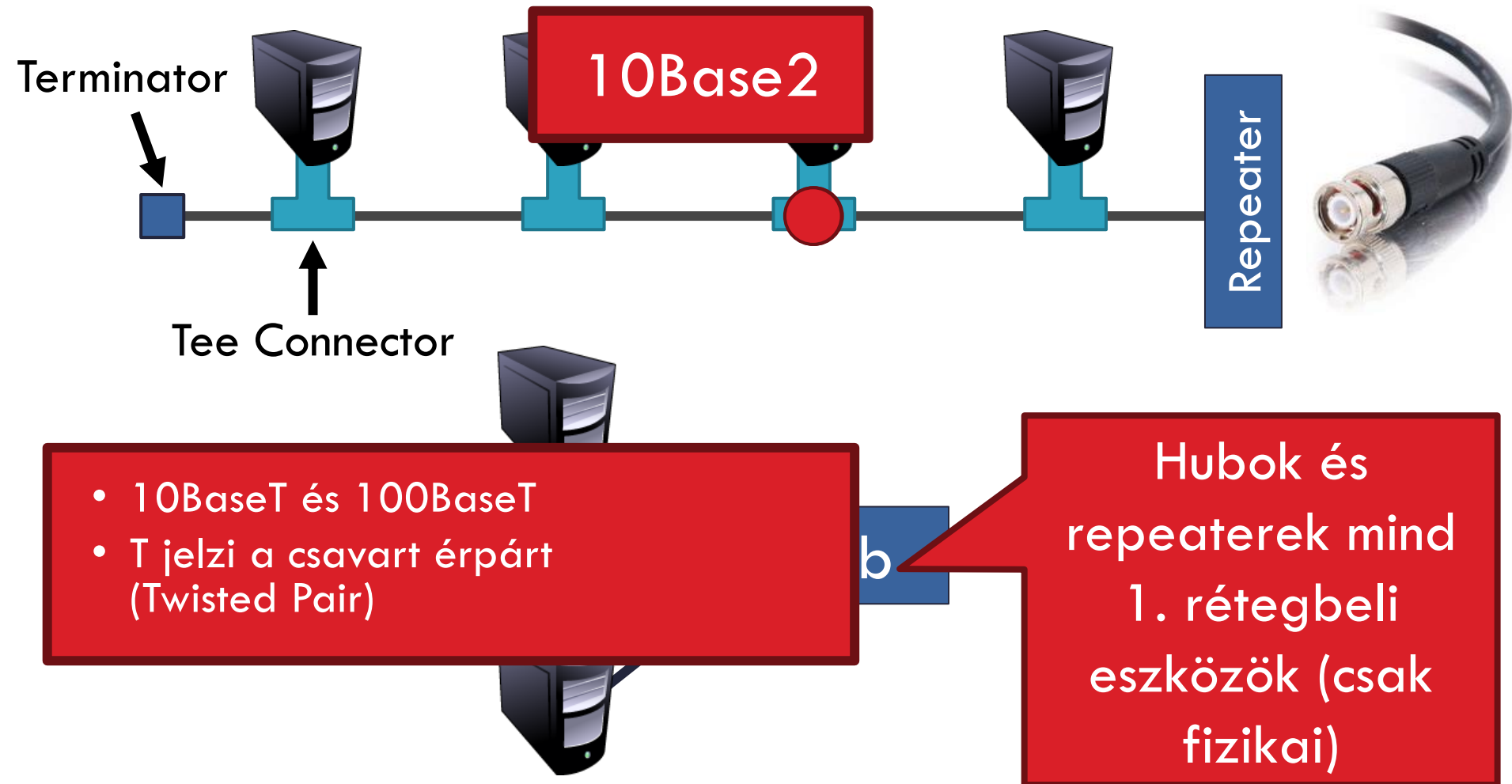
általán

órával

# Adatszóró (Broadcast) Ethernet

5

□ 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

7

- 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

8

- 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

9

- 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

10

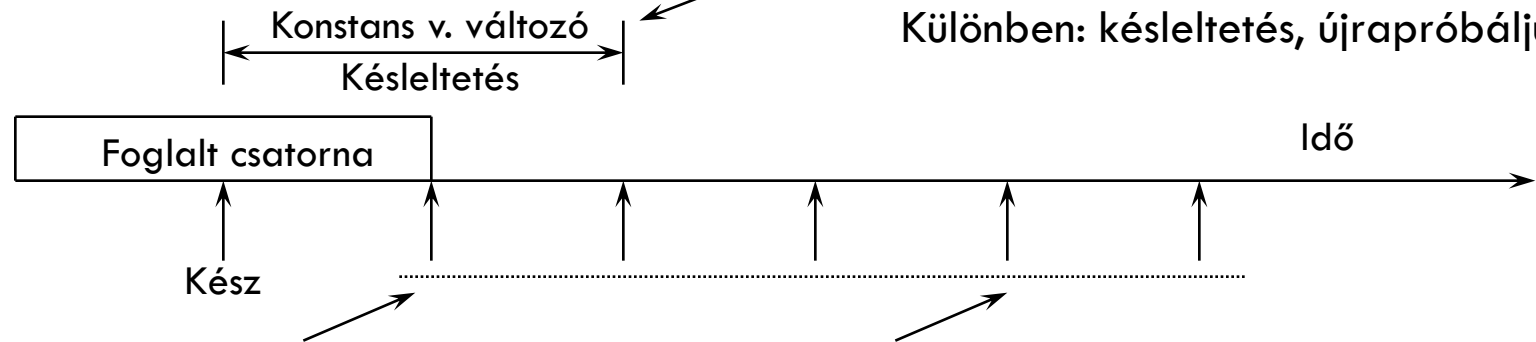
- Nem-perzisztens
- 1-perzisztens
- $p$ -perzisztens

## CSMA perzisztencia

**Nem-perzisztens:**

## Átvitel ha szabad

## Különb: késleltetés, újrapróbáljuk



**1-perzisztens:**

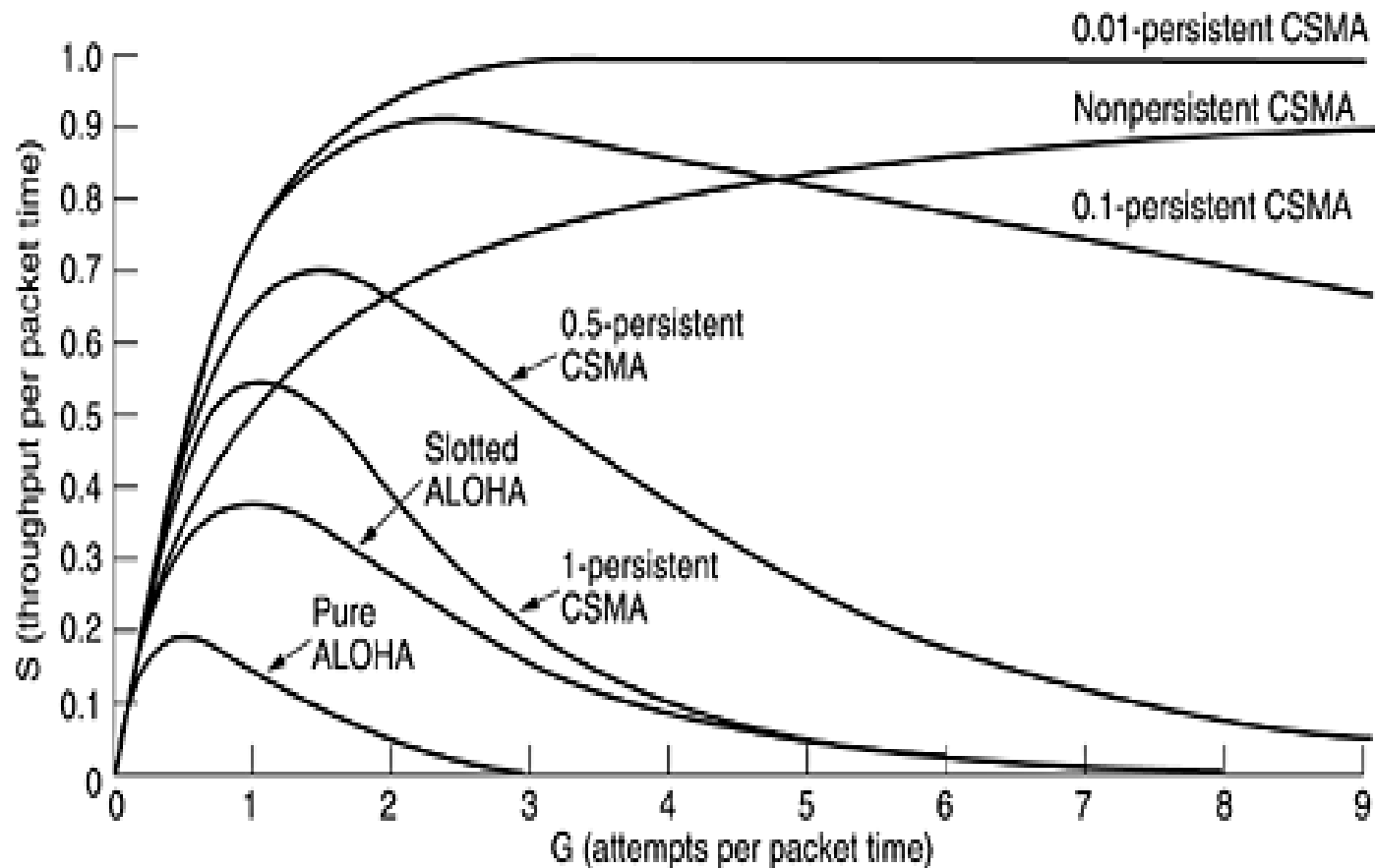
Átvitel amint a csatorna szabad  
Ütközés esetén visszalépés,  
majd újrapróbáljuk

**p-perzisztens:**

Átvitel p valószínűséggel, ha a csatorna szabad  
Különben: várunk 1 időegységet és újrapróbáljuk

# CSMA és ALOHA protokollok összehasonlítása

11



Forrás: [1]

# 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

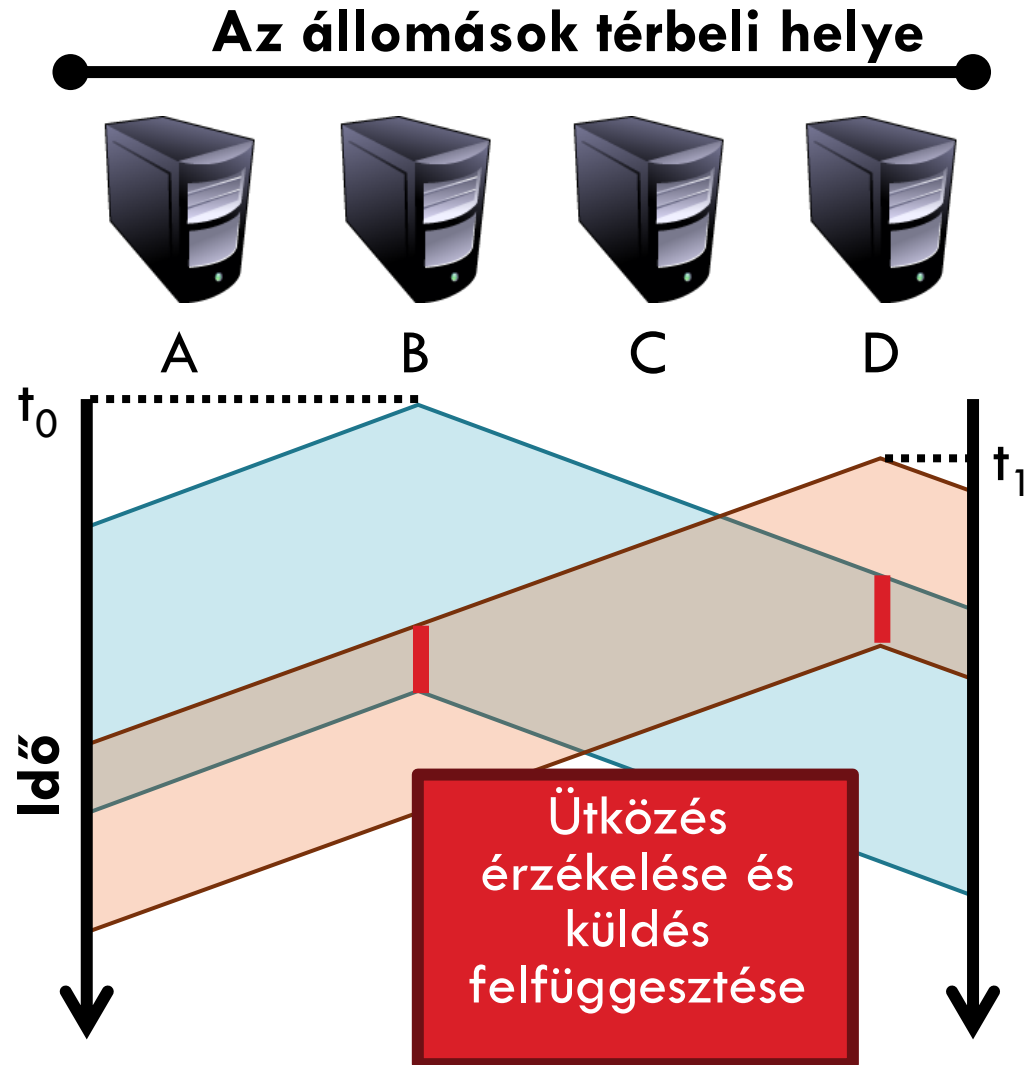
14

- ❑ 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

15

- 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

16

- 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

17

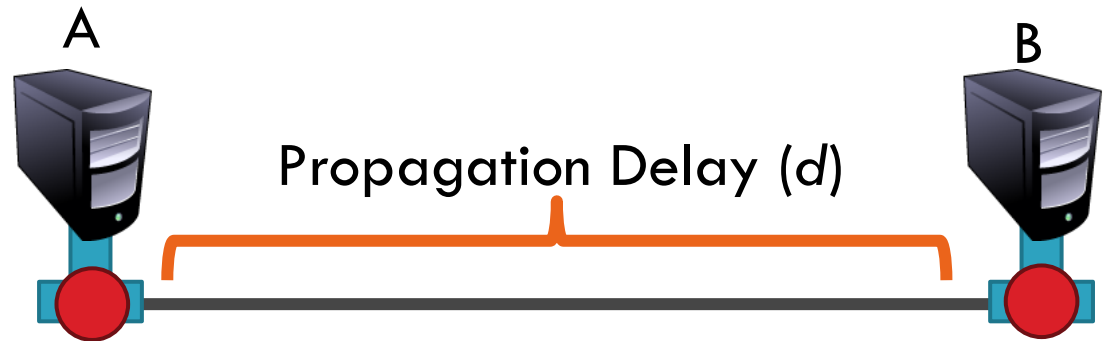
Tekintsünk két állomást, melyek üzenetei ütköztek

- 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

18

- 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



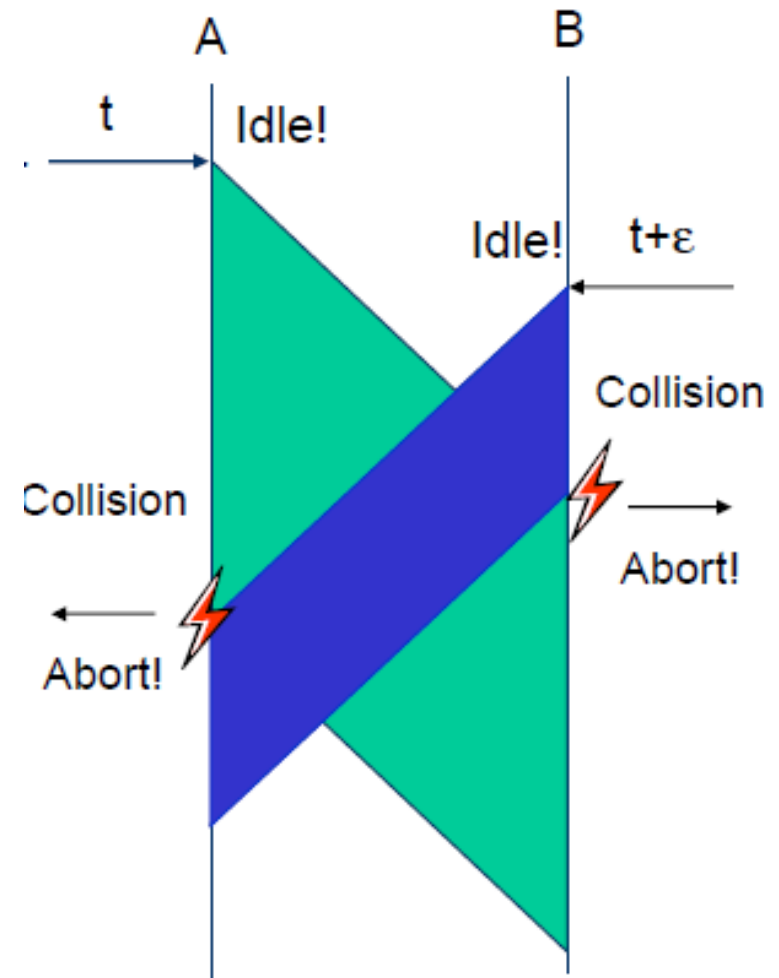
Alapötlet: Az A állomásnak  $2*d$  ideig kell küldenie!

# 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

20

- 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

21

- 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

22

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

- 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

23

- ❑ 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

24

## 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.



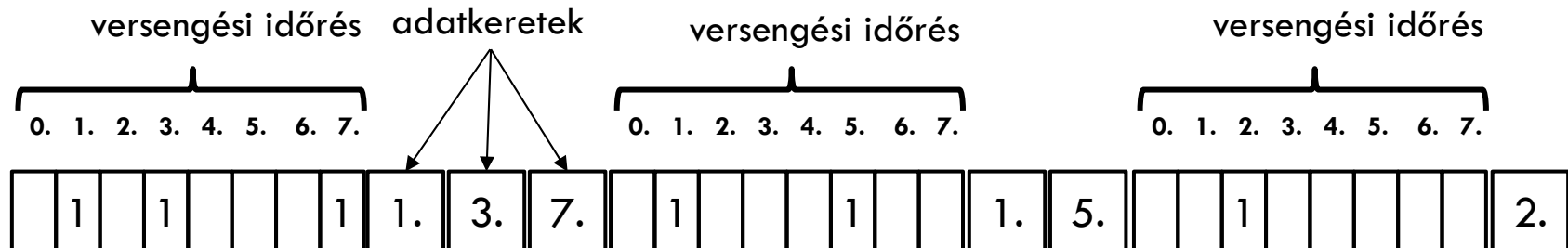
# Egy helyfoglalásos protokoll

25

- alapvető bittérkép eljárás

## MŰKÖDÉS

- Az ütköztetési periódus  $N$  időrés
- Ha az  $i$ -edik állomás küldeni szeretne, akkor a  $i$ -edik versengési időrásben 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

26

- 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

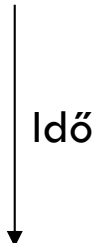
27

- **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

28

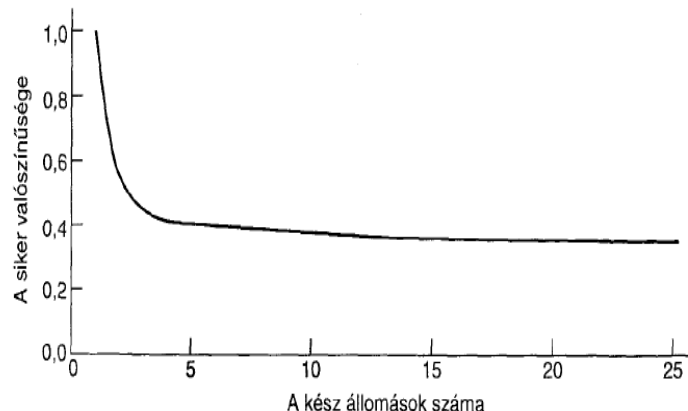
- **Cél:** Ötvözni a versenyhelyzetes és ütközésmentes protokollok jó tulajdonságait.
- **korlátozott versenyes protokoll** – Olyan protokoll, amely kis terhelés esetén versenyhelyzetes technikát használ a kis késleltetés érdekében, illetve nagy terhelés mellett ütközésmentes technikát alkalmaz a csatorna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

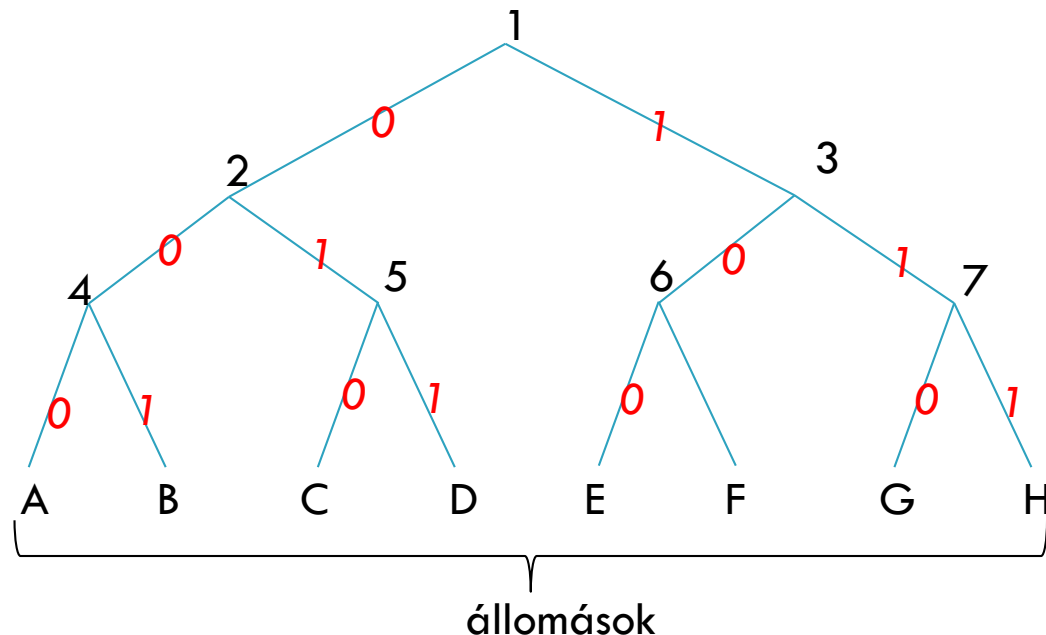


# Adaptív fabejárési protokoll 1 / 2

29

## Történeti háttér

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



# Adaptív fabejárási protokoll 2/2

30

## Működés

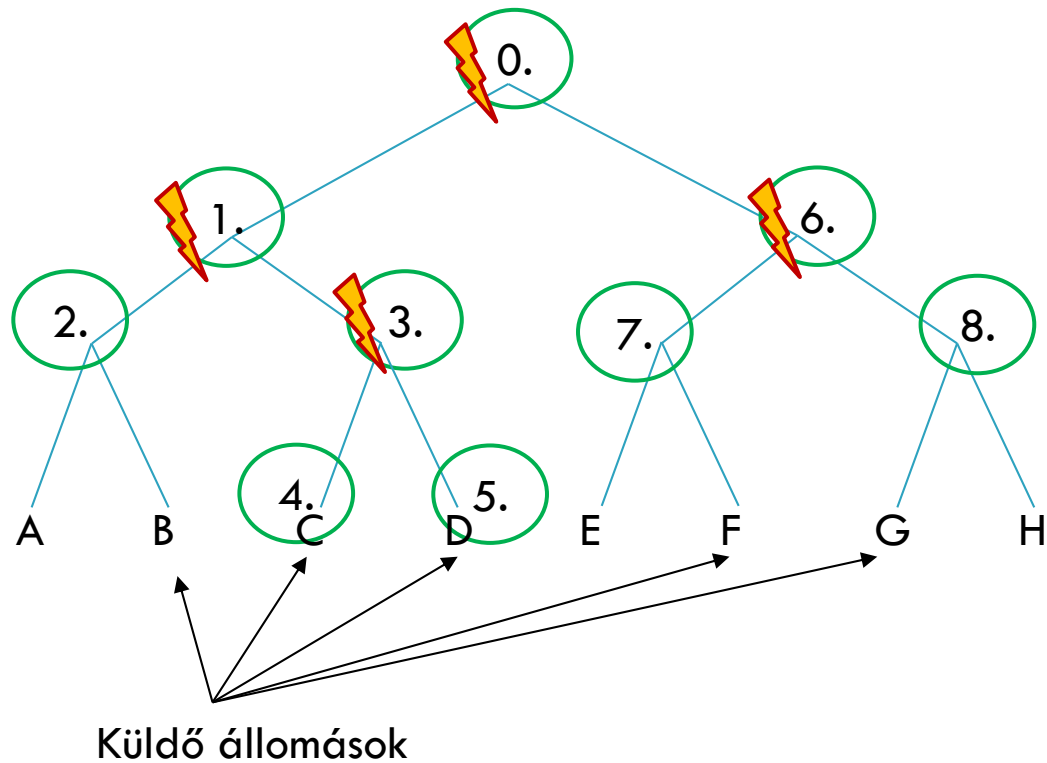
- 0-adik időrésben mindenki küldhet.
  - ▣ Ha ütközés történik, akkor megkezdődik a fa *mélységi bejárása*.
- A rések a fa egyes csomópontjaihoz vannak rendelve.
- Ütközéskor rekurzívan az adott csomópont bal illetve jobb gyerekcsomópontjánál folytatódik a keresés.
- Ha egy bitrés kihasználatlan marad,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

31



# Az adatkapcsolati réteg „legtetején”...

32



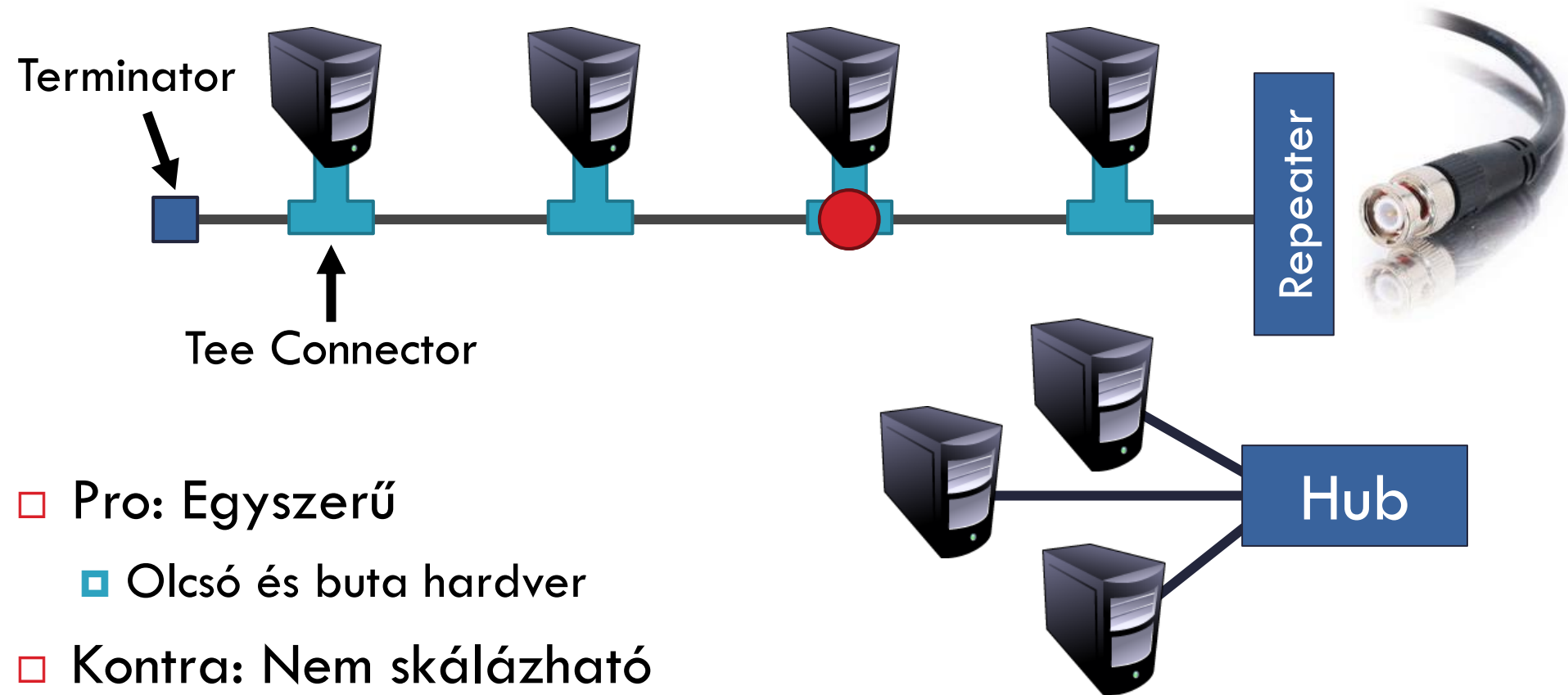
- Bridging, avagy hidak
  - ▣ Hogyan kapcsoljunk össze LAN-okat?
- Funkciók:
  - ▣ Keretek forgalomirányítása a LAN-ok között
- Kihívások:
  - ▣ Plug-and-play, önmagát konfiguráló
  - ▣ Esetleges hurkok feloldása



# Visszatekintés

33

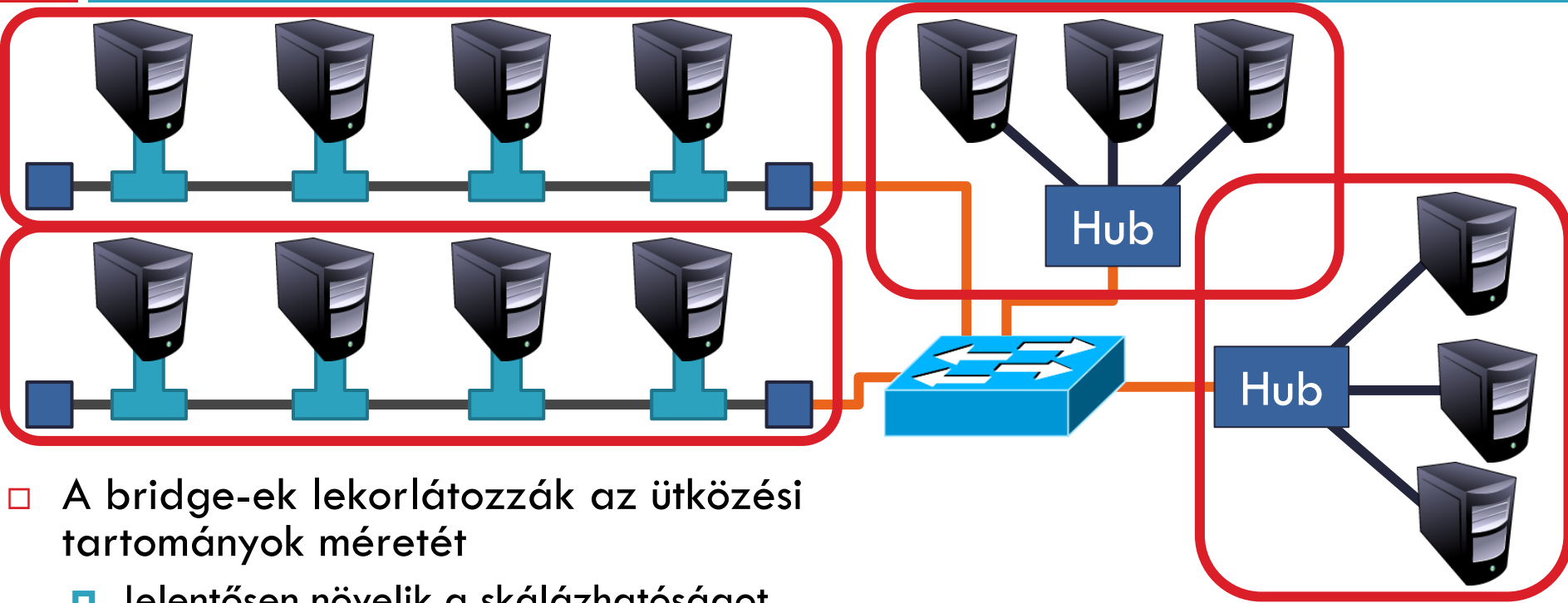
- Az Ethernet eredetileg adatszóró technológia volt



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

# LAN-ok összekapcsolása

34

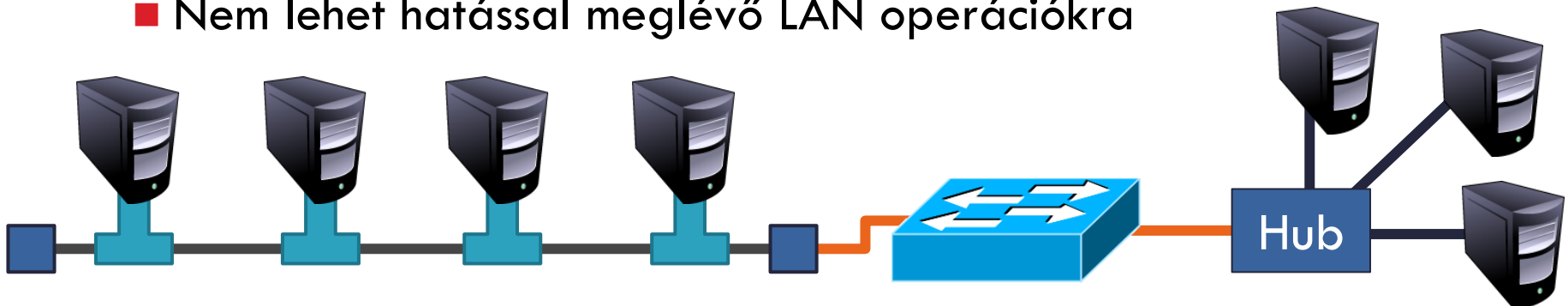


- A bridge-ek lekorlátozzák az ütközési tartományok méretét
  - ▣ Jelentősen növelik a skálázhatóságot
  - ▣ Kérdés: lehetne-e az egész Internet egy bridge-ekkel összekötött tartomány?
- Hátrány: a bridge-ek sokkal komplexebb eszközök a hub-oknál
  - ▣ Fizikai réteg VS Adatkapcsolati réteg
  - ▣ Memória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

# Bridge-ek (magyarul: hidak)

35

- ❑ 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)

36

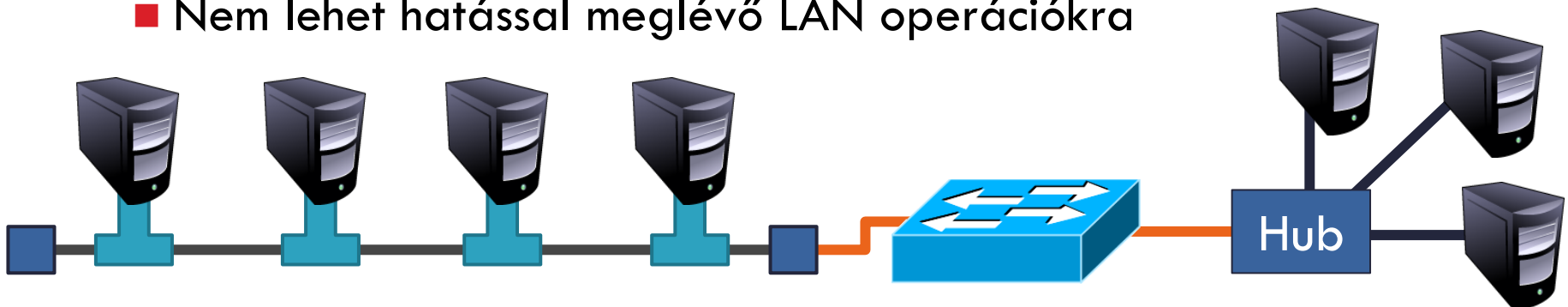
- Az Ethernet switch eredeti formája

□

□

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

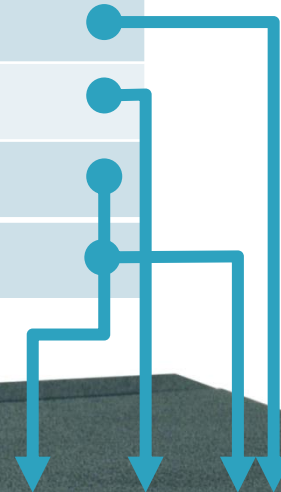


# Keret Továbbító Táblák

37

- 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



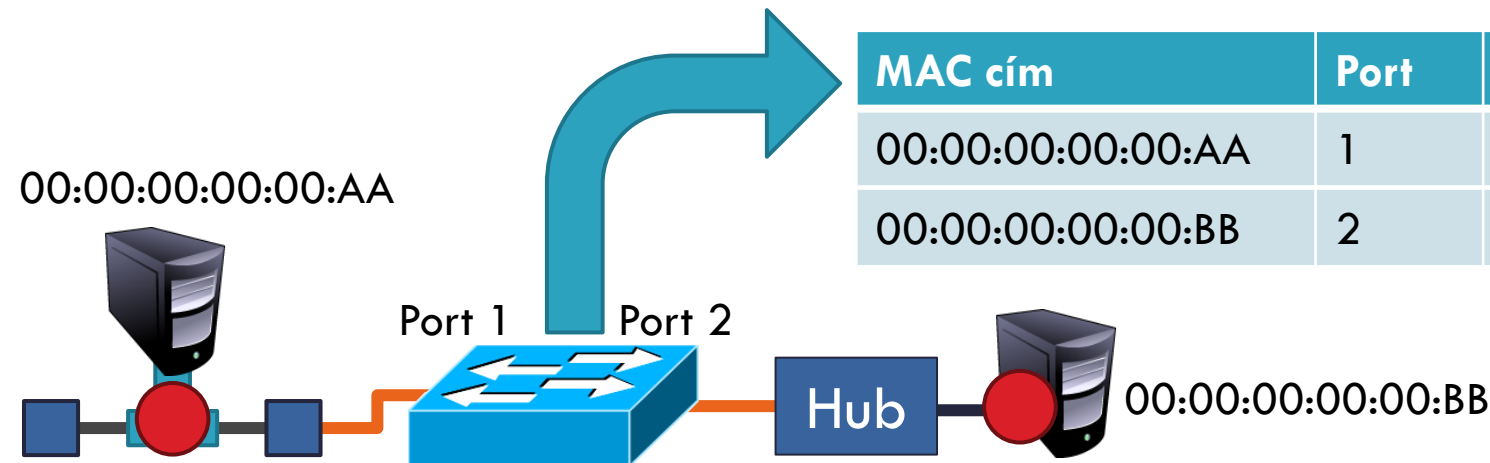
# Címek tanulása

38

- ❑ 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

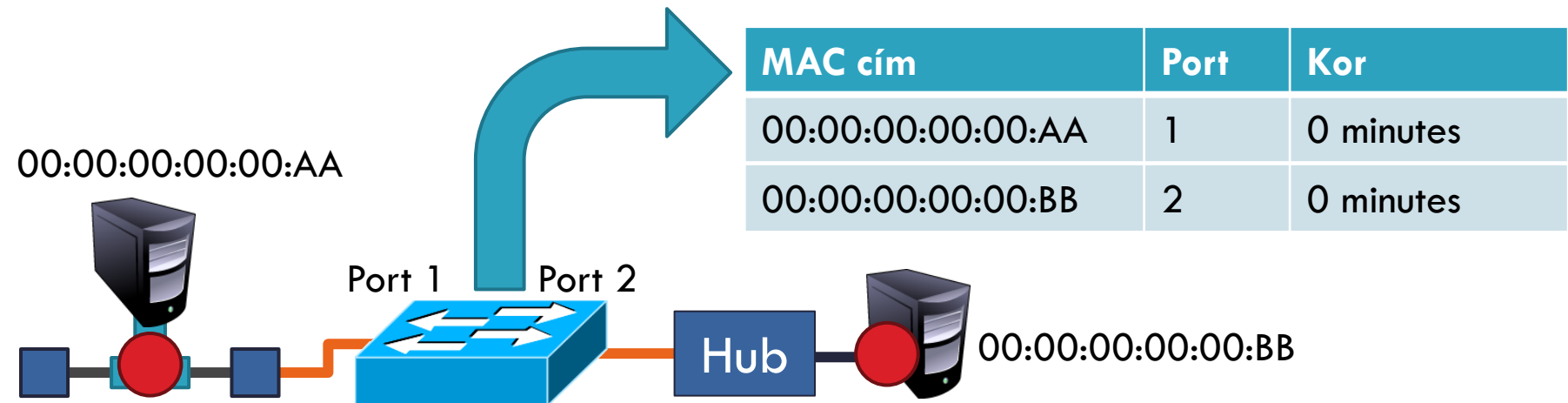
MAC cím	Port	Kor
00:00:00:00:00:AA	1	0 minutes
00:00:00:00:00:BB	2	0 minutes



# Címek tanulása

39

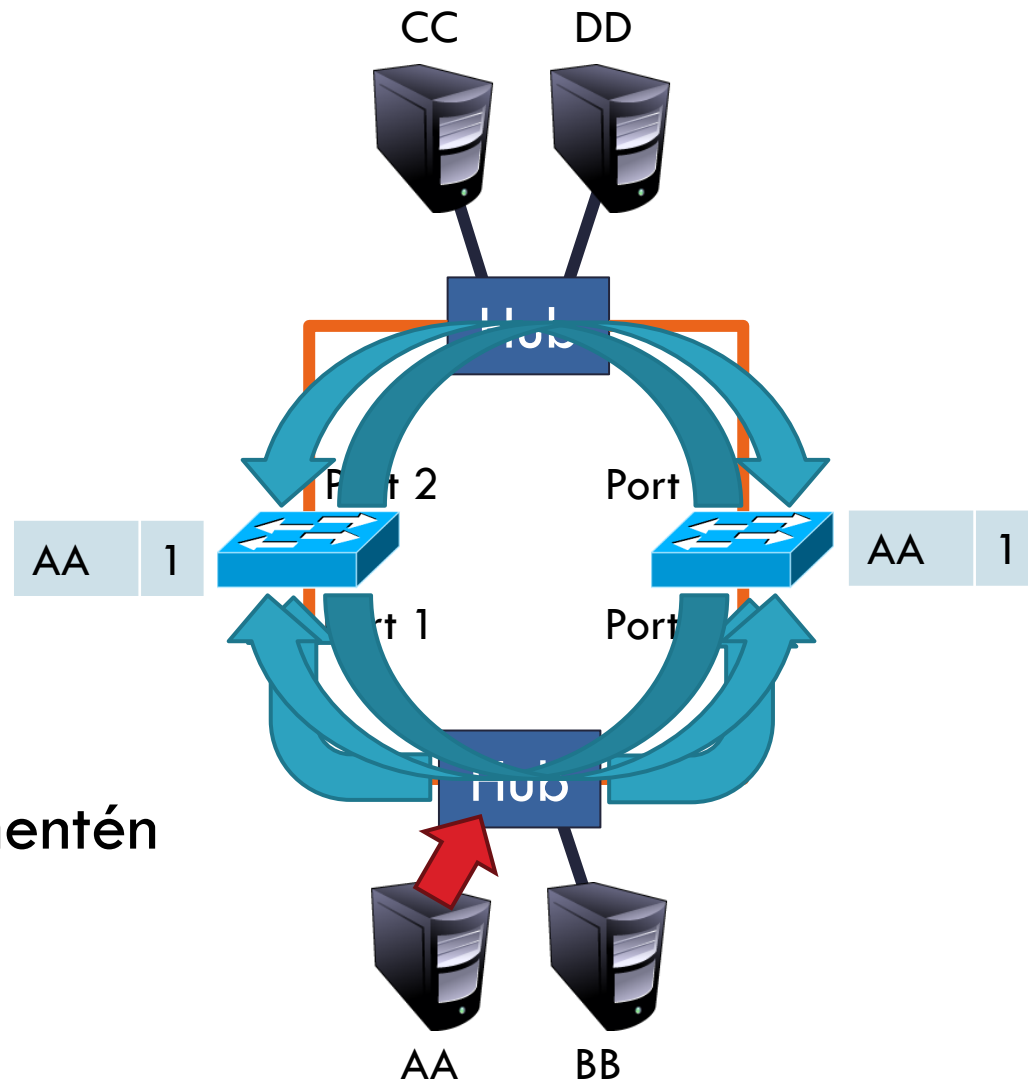
- ❑ Kézi beállítás is lehetséges, de...
  - ▣ Időigényes
  - ▣ Potenciális hiba forrás
  - ▣ Nem alkalmazkodik a változásokhoz (új hosztok léphetnek be és régiek hagyhatják el a hálózatot)
- ❑ Ehelyett: tanuljuk meg a címeket
  - ▣ Tekintsük a **forrás címeket** a különböző portokon beérkező kereteknek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



# Hurkok problémája

40

- ❑  $\langle \text{Src}=\text{AA}, \text{Dest}=\text{DD} \rangle$
- ❑ 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

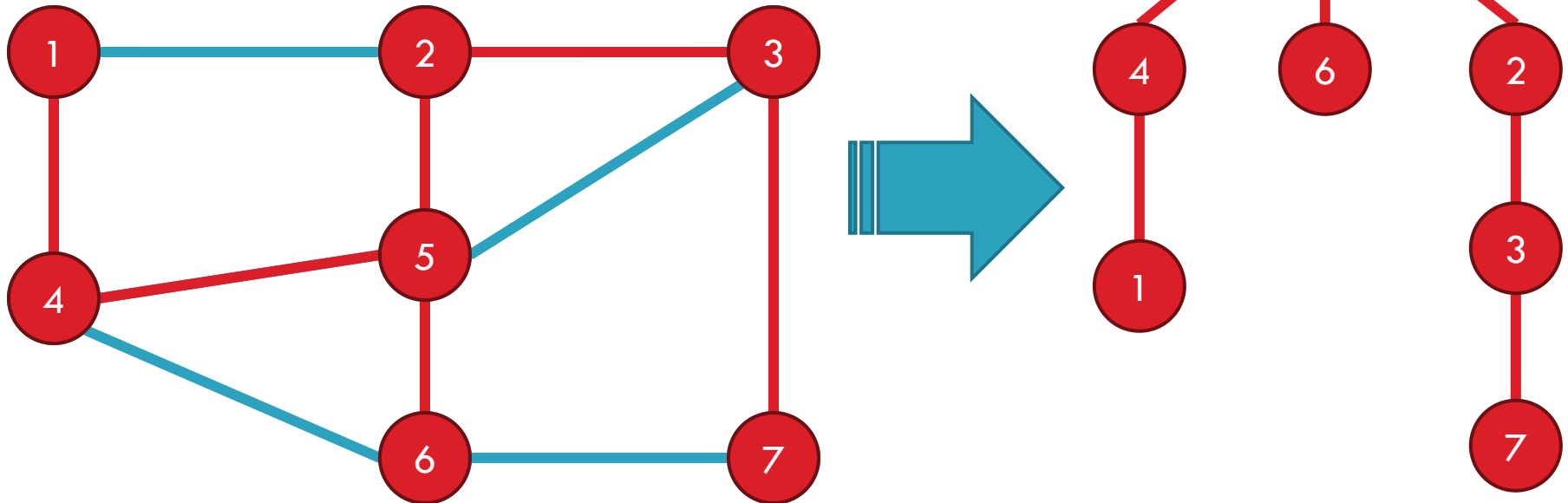




# Feszítőfa

41

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



# A 802.1 feszítőfa algoritmus

42

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

43

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

Bridge ID

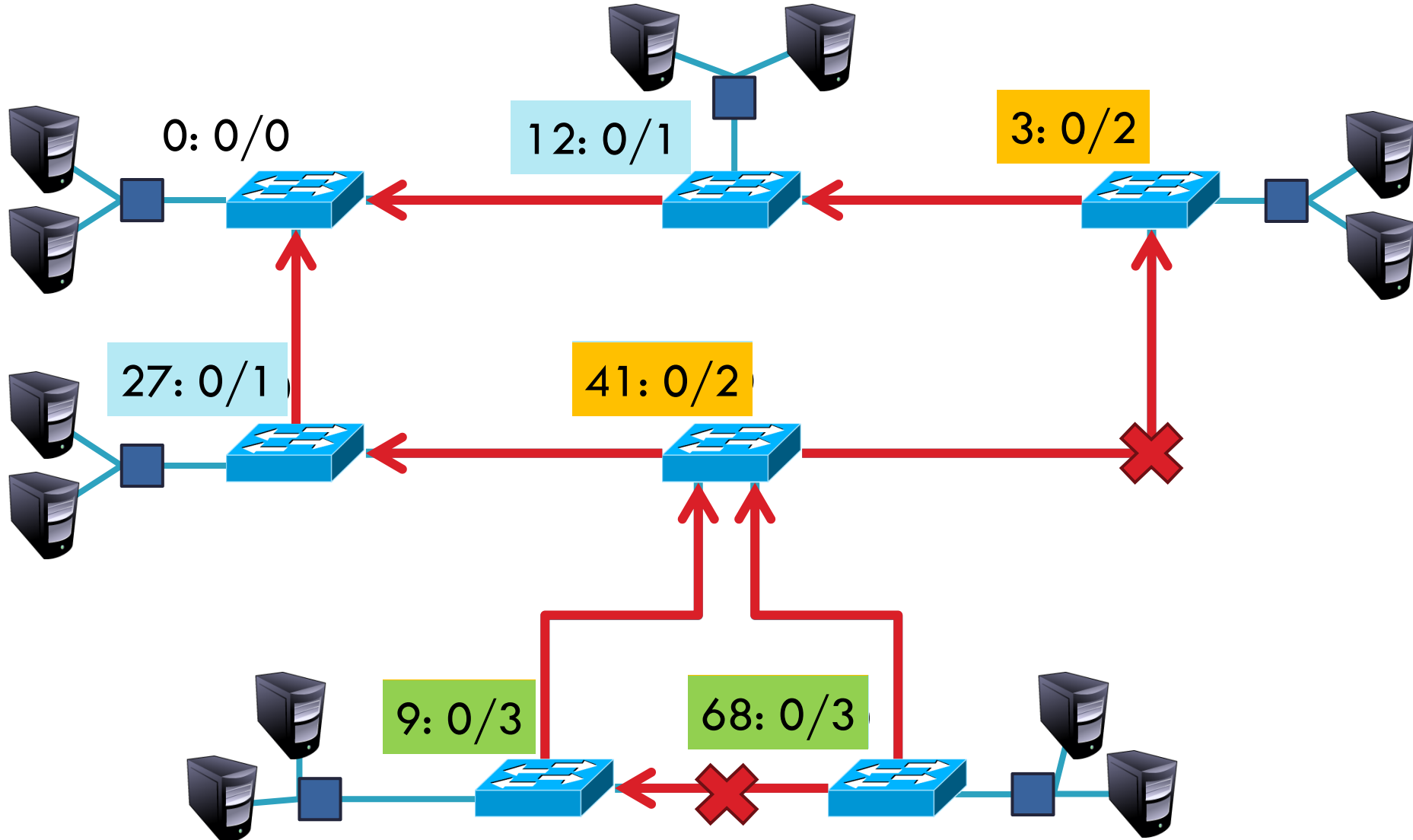
Gyökér ID

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

44



# Bridge-ek vs. Switch-ek

## Hidak vs. Kapcsolók

45

- ❑ 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

46

- ❑ 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

47

- ❑ 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!