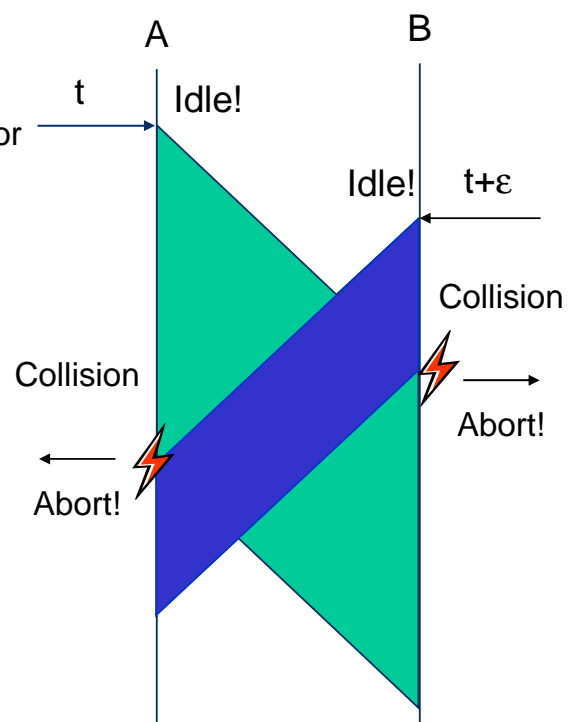


Számítógépes Hálózatok 2013

6. Adatkapcsolati réteg, MAC – CSMA/CD, versenymentes protokollok, korlátozott verseny, Ethernet; LAN-ok összekapcsolása

Kollízió felismerés (collision detection) – CSMA/CD

- Ha két csomag ütközik, sok idő veszik el azok átvitelének befejezésére
 - Ha lehetséges lenne felismerni egy kollíziót amikor az fellép, az átvitelt lehetne abortálni és egy új próbát tenni
 - Az elvesztegetett idő csökken, nem kell megvárni, hogy a (szétrombolt) csomagok befejeződjenek
 - A fizikai rétegtől függően, a kollízió felismerhető!
 - Szükséges: A küldőnek képesnek kell lenni „hallgatni” a médiumot miközben küld és összehasonlítani amit küld és amit „hall”
 - Ha különbözik: Kollízió
- **CSMA/CD – Carrier Sense Multiple Access/Collision Detection**
- Feltétel, hogy felismerjük mindkét oldalon:
 $T_{gen} \geq 2d$
 - T_{gen} : csomag generálási ideje

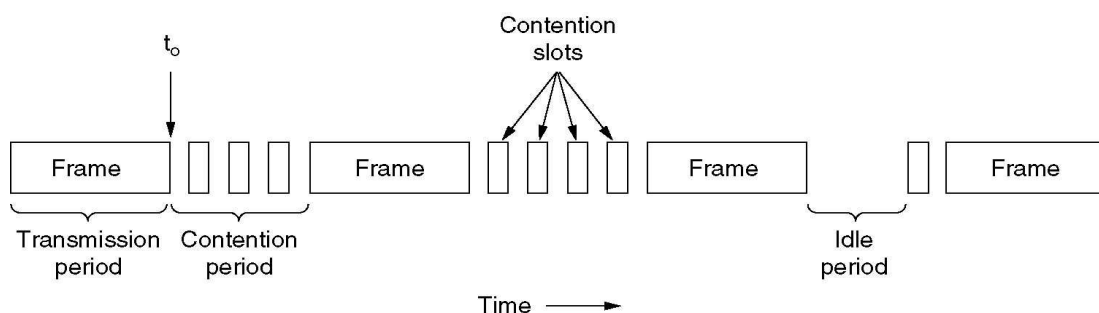


Mi a teendő kollízió esetén?

- Az állomások át akarják vinni a csomagjaikat a kollízió ellenére
 - Újra meg kell próbálniuk
 - Azonnal? Ez egy másik kollíziót okozna
 - Valahogy koordinálva? Nehéz, nem áll rendelkezésre kommunikációs médium
 - Várjunk egy véletlen ideig!
 - Randomizálás “deszinkronizálja” a médium hozzáférést, és ezzel segít elkerülni a kollíziót
 - Valamennyi kihasználatlan időt eredményez
- Váltakozva verseny- és átviteli-periódusok

CSMA/CD periódusai

- Üres periódus (IDLE)
 - Egyik állomás sem küld frame-et
 - Verseny periódus (Contention Period)
 - Kollíziók történhetnek, az átvitel abortálódik
 - Átviteli periódus (Transmission Period)
 - Nincs Kollízió, a protokoll effektív része
- Csak verseny-, átviteli- és üres periódus van



Hogy válasszuk meg a véletlen várakozási időt?

- A legegyszerűbb választás: Válasszunk ki egyet k slot közül
 - Egyszerűség kedvéért tételezzünk fel egy slot-okra osztott idő modellt
 - Egyenletes eloszlás szerint $\{0, \dots, k-1\}$ felett
 $[0, \dots, k-1]$: verseny ablak (**contention window**)
 - Kérdés: hogy válasszunk meg k -t?
 - Kicsi k : Kicsi delay, de nagy az esély ismételt kollízióra
 - Nagy k : Kicsi az ismételt kollízió esélye (mivel az állomások kísérletei egy nagy intervallumra oszlanak el), de szükségtelenül nagy a delay, ha csak kevés állomás akarja használni a csatornát
- **Adaptáljuk** k választásához az állomások aktuális számát / csatorna terhelést

Hogyan változtassuk k -t a terheléstől függően?

- Egy lehetőség: derítsük ki *valahogy* explicit az állomások számát, számítsunk ki ehhez egy optimális k -t, tudassuk ezt minden állomással
 - Nehéz, magas overhead, ...
 - Lehetséges egy *implicit* megoldás?
- Milyen következményekkel jár egy kicsi k , ha a terhelés nagy?
 - Sok kollízió!
 - Tehát: Használjuk a kollíziókat indikátorként, hogy a verseny ablak túl kicsi – növeljük meg a verseny ablak méretét!
 - Csökkenti a kollíziók valószínűségét, automatikusan adaptálja a terhelés növekedését
- Kérdés: Hogy növeljük k -t a kollízió után, hogy csökkentsük újra?

Hogy változtassuk k-t – Binary exponential backoff

- Növeljük k-t a **kollízió után**: sok lehetőség van
 - Általánosan használt: **duplázzuk** meg k-t
 - De csak egy korlátig, mondjuk, 1024 slot – kezdjük k=2-vel
 - Ezt a stratégiát **binary exponential backoff**-nak hívják
- Csökkentsük k-t, ha elegendően sok frame kollízió mentesen átvitelre került
 - Lehetőségek: vonjunk ki belőle egy konstans, felezzük meg, ...
 - Viszonylag komplikált, erőforrást pazarolhat, miközben nem elég agilis
 - A legegyszerűbb: induljunk megint k=1-gyel
 - Általánosan használt

Hogy változtassuk k-t – Binary exponential backoff

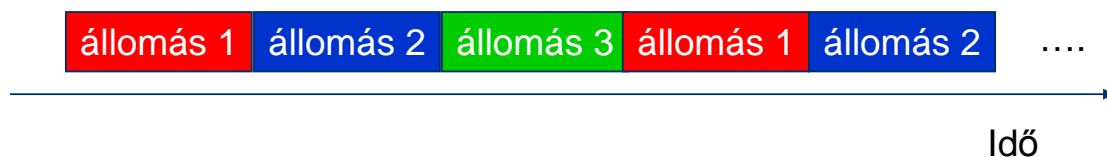
- Algoritmus **binary exponential backoff**
 - $k := 2$
 - Amíg az utolsó küldésnél kollízió történt
 - Válasszuk i-t egyenlő valószínűséggel véletlenül $\{0, \dots, k-1\}$ közül
 - Várjunk i slot-ot
 - Küldjük a frame-et (kollízió felismerése esetén: abort)
 - Ha $k < \text{limit}$: $k := 2k$
- Ez az algoritmus
 - a várakozási időt dinamikusan a csatornát használó állomások számához igazítja
 - gondoskodik a csatorna egyenletes kihasználásáról
 - fair (hosszú távon)

MAC alréteg

- Statikus Multiplexálás
- Dinamikus csatorna foglalás
 - Kollízió alapú protokollok
 - **Verseny-mentes protokollok (contention-free)**
 - Protokollok korlátozott versennyel (limited contention)
- Az Ethernet példája

Verseny mentes protokollok

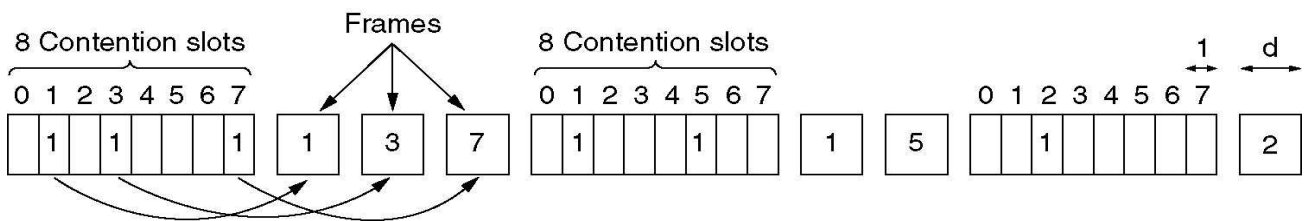
- Egyszerű példa: Statikus (idő-) multiplexálás (TDMA)
 - Minden állomáshoz egy fix idő-slotot rendelünk egy ismétlődő idő idő-séma szerint



- Hátrányait elemeztük
- Van-e dinamikus kollízió mentes protokoll?

Bit-map protokoll

- A TDMA problémája
 - Ha az állomás nem küld semmit, az idő-slotja kihasználatlan
- Foglalási rendszer: Bit-map protocol
 - Rövid statikus foglalás-slotok, melyek jelzik az átvitel kívánságot
 - Minden állomásnak hallani kell



Bitmap-Protokollok

- Tulajdonságok alacsony terhelés esetén
 - Ha nincs csomagküldés, akkor az (üres) verseny-slot ismétlődik
 - Egy állomás, ha küldeni akar, meg kell várnia a verseny-slotokat
 - Viszonylag nagy késés (delay)
- Tulajdonságok nagy terhelés esetén
 - A csatornát az adatcsomagok dominálják
 - Az adatcsomagok nagyobbak mint a verseny-slotok
 - Az overhead elhanyagolható
 - Jó és stabil átvitel (throughput)
- Bitmap egy Carrier-Sense protokoll!

MAC alréteg

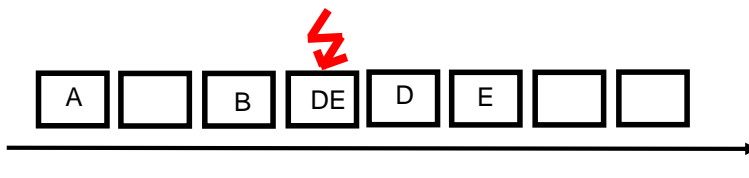
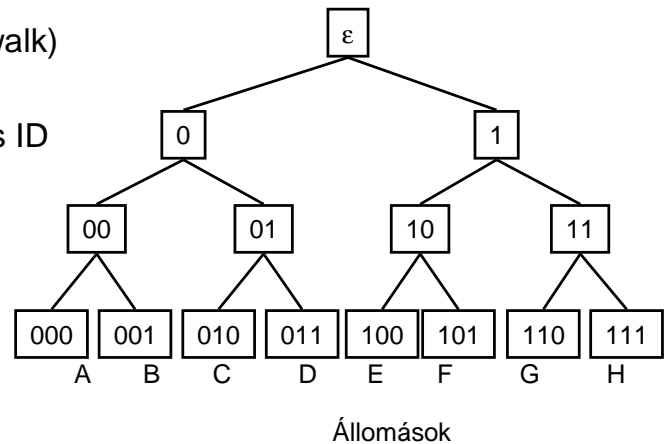
- Statikus Multiplexálás
- Dinamikus csatorna foglalás
 - Kollízió alapú protokollok
 - Verseny-mentes protokollok (contention-free)
 - **Protokollok korlátozott versennyel (limited contention)**
- Az Ethernet példája

Protokollok korlátozott versennyel

- Cél:
 - kis késés (delay) alacsony terhelés esetén
 - mint a kollízió alapú protokolloknál
 - nagy átvitel (throughput) nagy terhelés esetén
 - mint a verseny mentes protokolloknál
- → korlátozott verseny (verseny a verseny slotoknál)
- Ötlet:
 - A verseny slotokhoz vegyük figyelembe a résztvevő állomások számát
 - Több állomásnak kell használni egy slotot

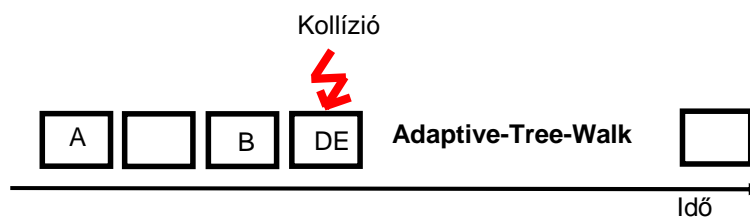
Adaptív fa bejárás protokoll

- Adaptív fa bejárás protokoll (adaptive tree walk)
- Kiindulópont:
 - Minden állomást egy egyértelmű, bináris ID reprezentál
 - Az ID-k egy fa leveleinek felelnek meg
 - Szinkronizált protokoll
 - A fa egy u csomópontjánál 3 esetet különböztethetünk meg:
 - nem küld állomás u részfájában
 - pontosan egy állomás küld
 - kollízió: legalább két állomás küld



Adaptív fa bejárás protokoll – Alap algoritmus

- Alap_Algoritmus
 - Minden állomás azonnal küld (slotted Aloha)
 - Ha kollízió lép fel,
 - Egy állomás sem fogad el új csomagot a hálózati rétegtől
 - Hajtsuk végre az **adaptive_tree_walk(ε)** eljárást



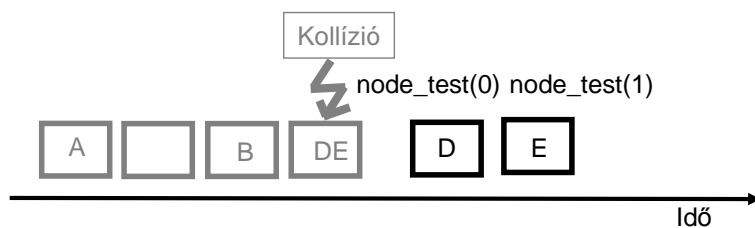
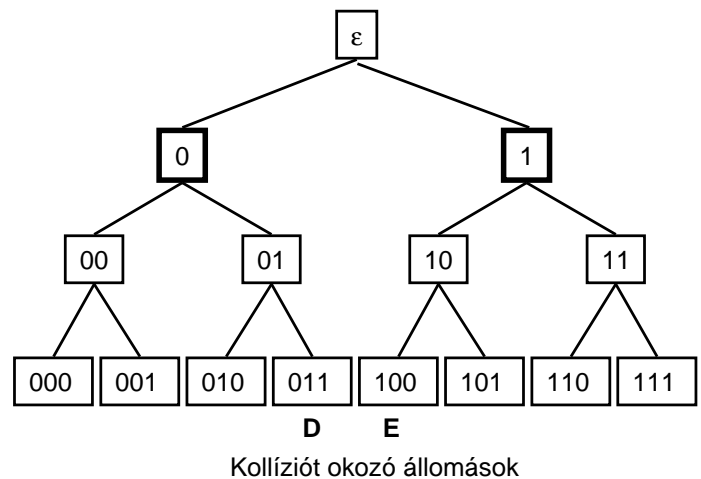
Adaptív fa bejárás protokoll – Csomópont teszt

- Csomópont-teszt algoritmus (node_test)

- a fa egy u csomópontjához

- node_test(u)

- Tekintsünk egy slotot u-hoz
- A slotban azon állomások küldenek, amelyek u részfájában vannak (amelyek ID-ja u-val kezdődik)



Adaptív fa bejárás protokoll – Alap algoritmus

- Csomópont-teszt algoritmus

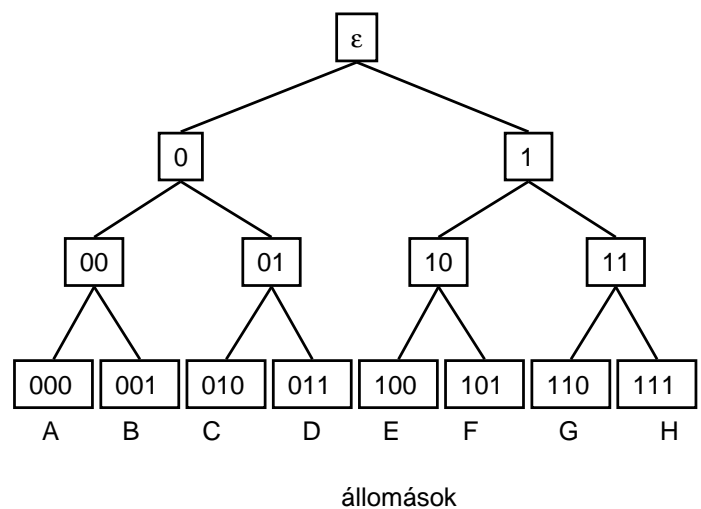
- a fa egy u csomópontjához

- node_test(u)

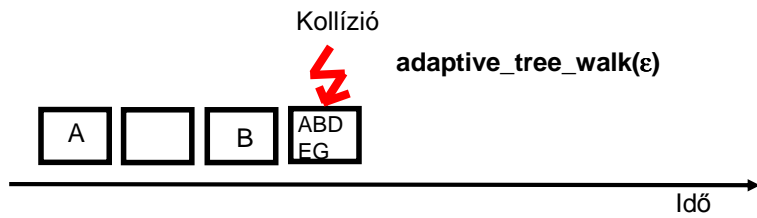
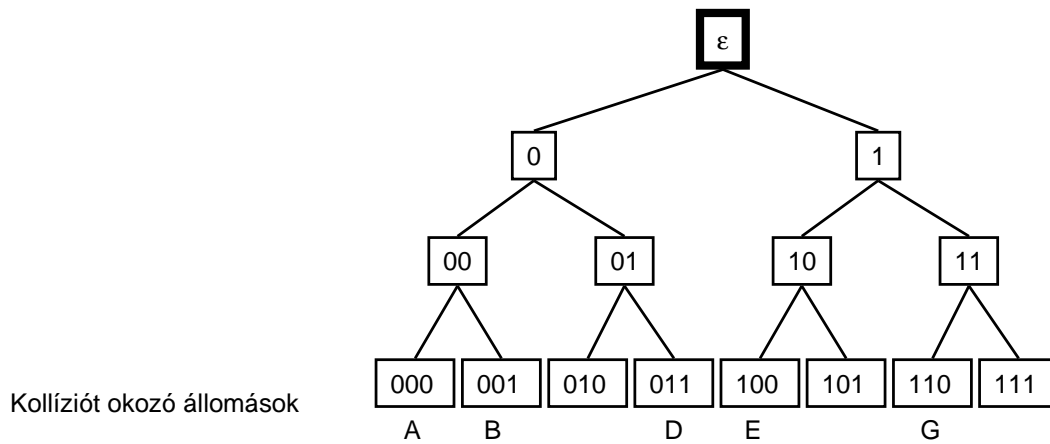
- Tekintsünk egy slotot a fa u csomópontjához
- A slotban azon állomások küldenek, amelyek u részfájában vannak (amelyek ID-je u-val kezdődik)

- adaptive_tree_walk(x)

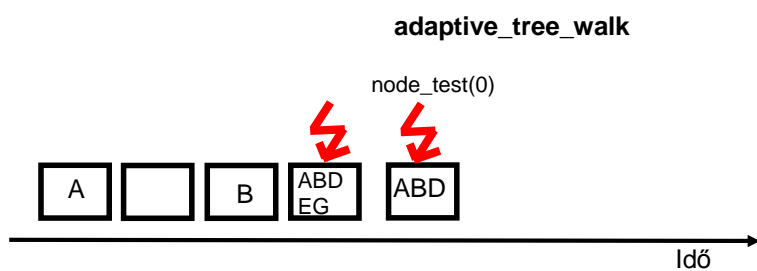
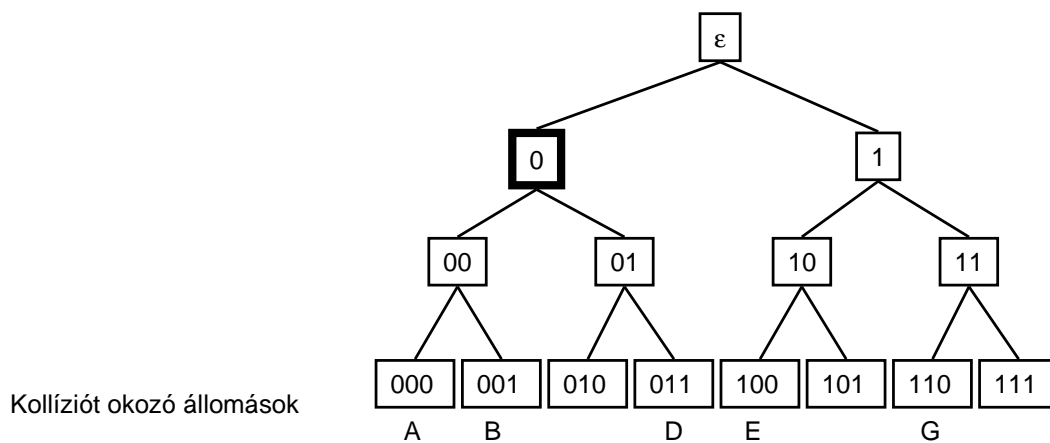
- node_test(x0)
- Ha kollízió lép fel,
 - adaptive_tree_walk(x0)
- node_test(x1)
- Ha kollízió lép fel,
 - adaptive_tree_walk(x1)



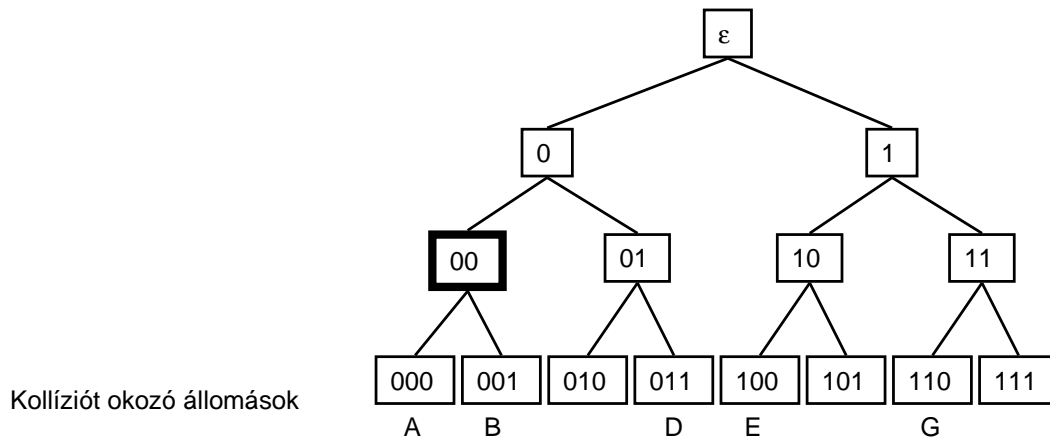
Adaptív fa bejárás protokoll – Példa (1)



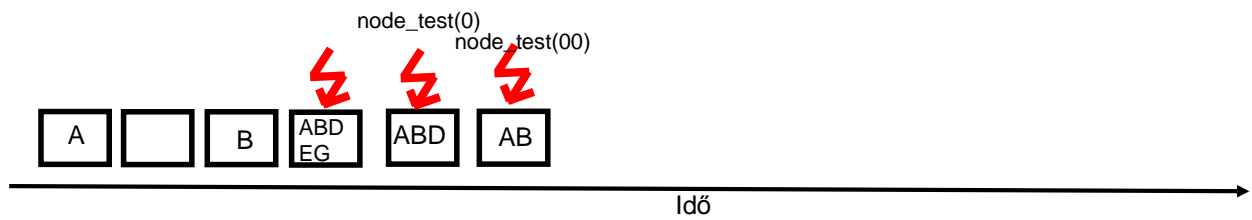
Adaptív fa bejárás protokoll – Példa (2)



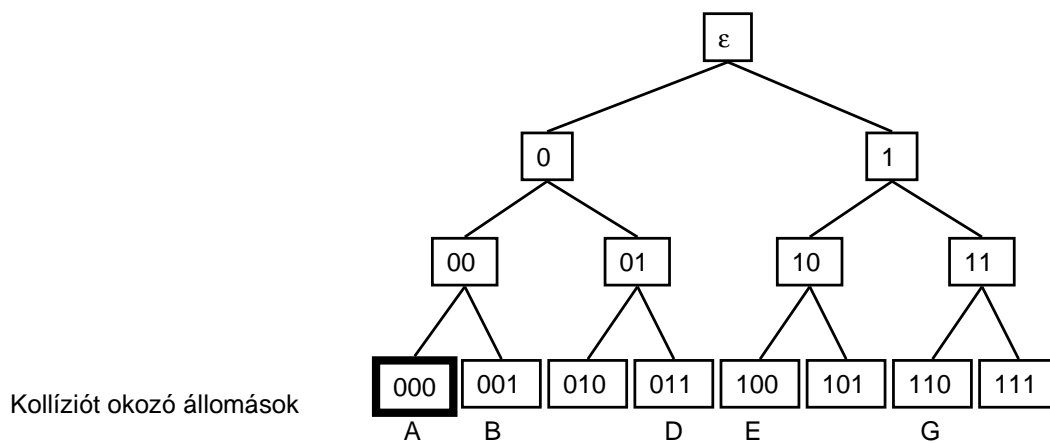
Adaptív fa bejárás protokoll – Példa (3)



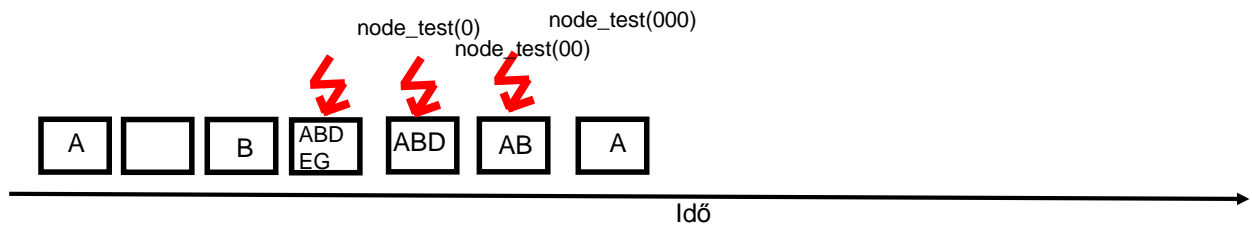
Adaptive_tree_walk



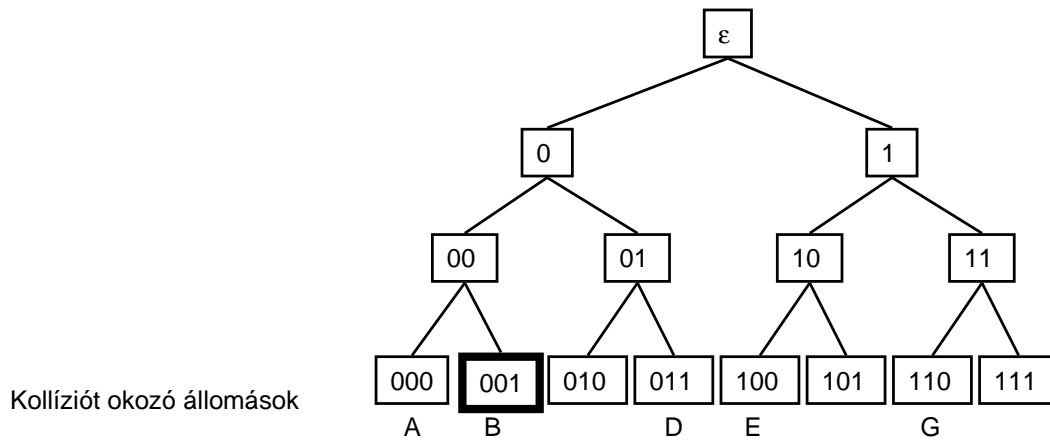
Adaptív fa bejárás protokoll – Példa (4)



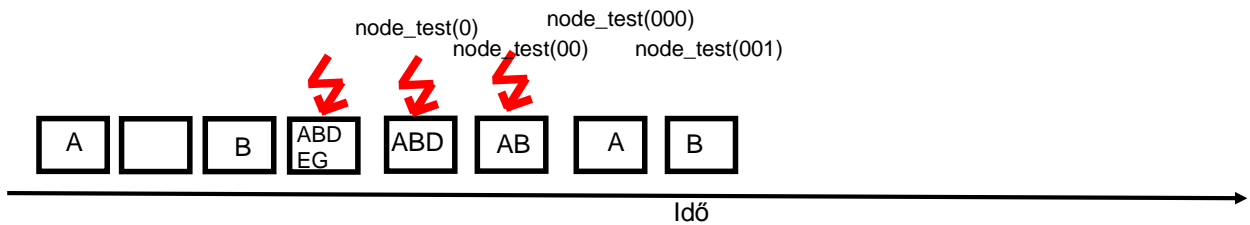
Adaptive_tree_walk



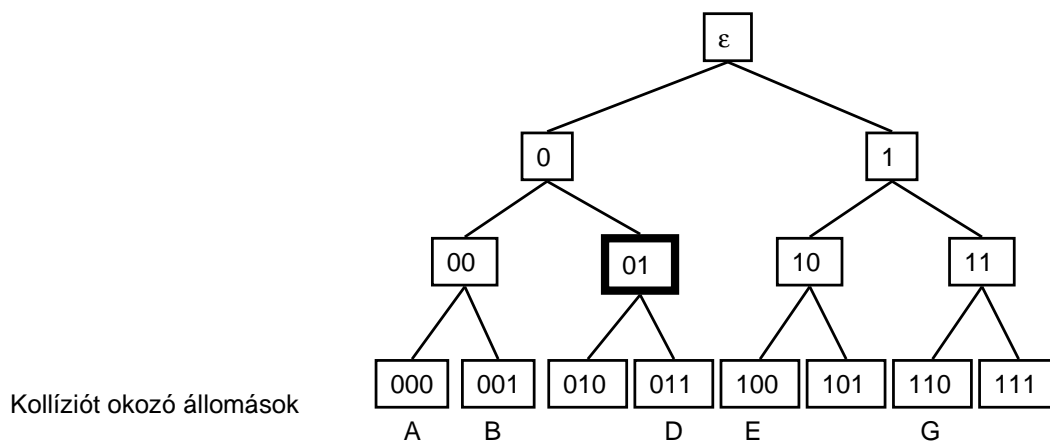
Adaptív fa bejárás protokoll – Példa (5)



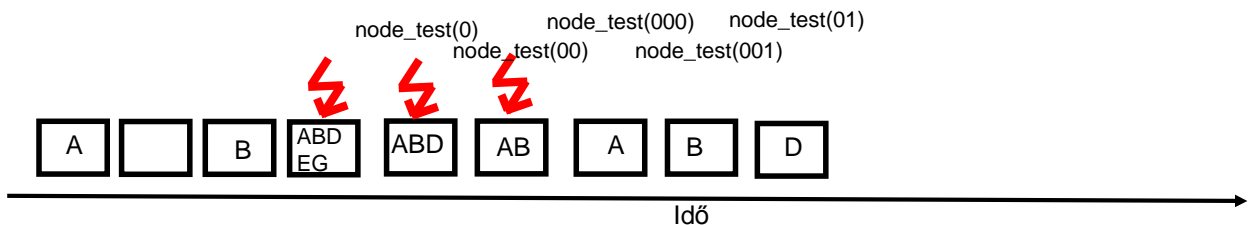
Adaptive_tree_walk



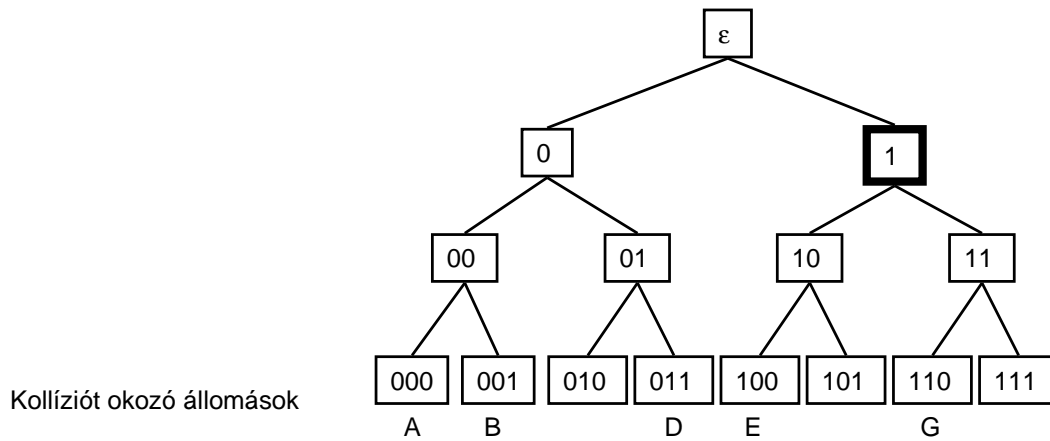
Adaptív fa bejárás protokoll – Példa (6)



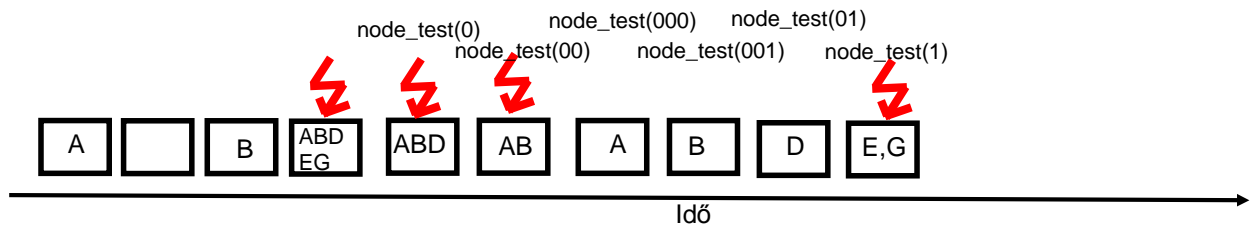
Adaptive_tree_walk



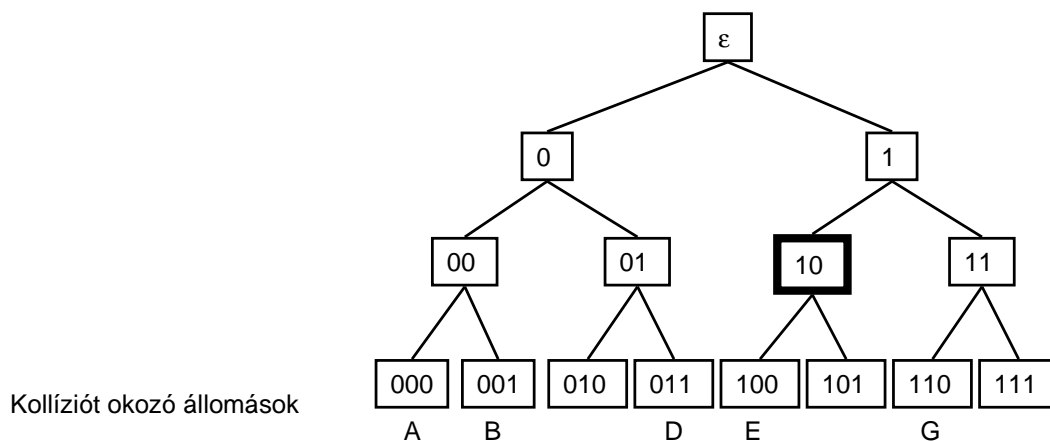
Adaptív fa bejárás protokoll – Példa (7)



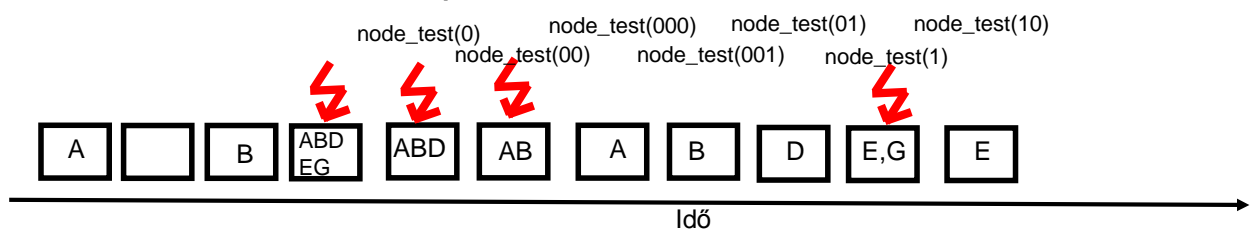
Adaptive_tree_walk



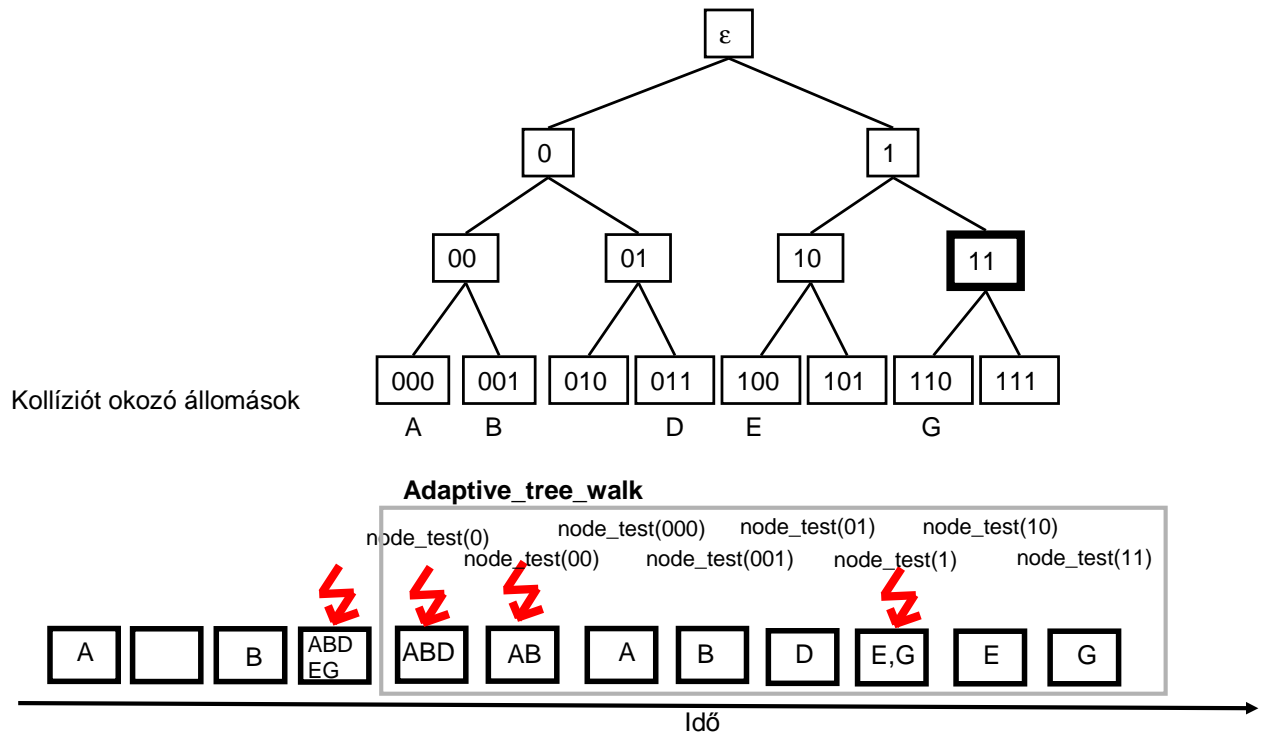
Adaptív fa bejárás protokoll – Példa (8)



Adaptive_tree_walk



Adaptív fa bejárás protokoll – Példa (9)



MAC alréteg

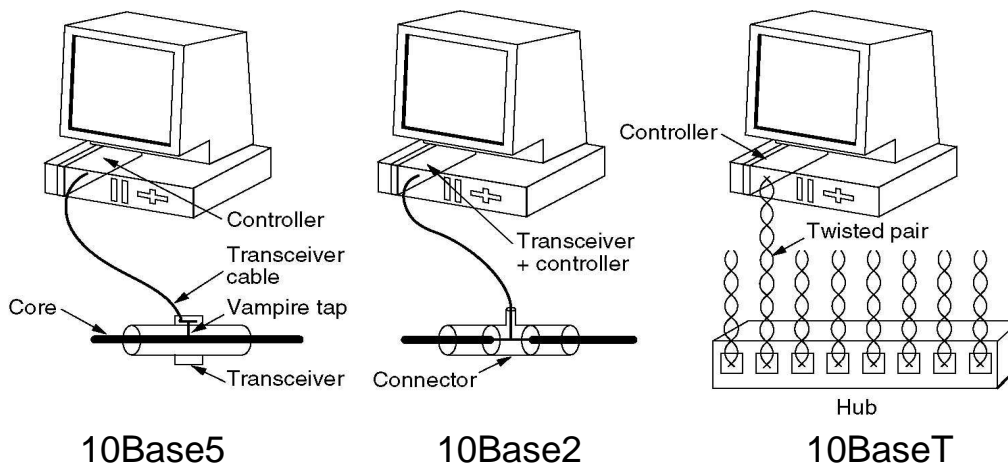
- Statikus Multiplexálás
- Dinamikus csatorna foglalás
 - Kollízió alapú protokollok
 - Verseny-mentes protokollok (contention-free)
 - Protokollok korlátozott versennyel (limited contention)
- **Az Ethernet példája**

Az Ethernet példája

- Gyakorlati példa: Ethernet
 - IEEE 802.3 standard
- A IEEE 802.3 standard pontjai
 - Vezeték
 - Fizikai réteg
 - Adatkapcsolati réteg médium hozzáférés kontrollal

Ethernet – Vezetékek

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

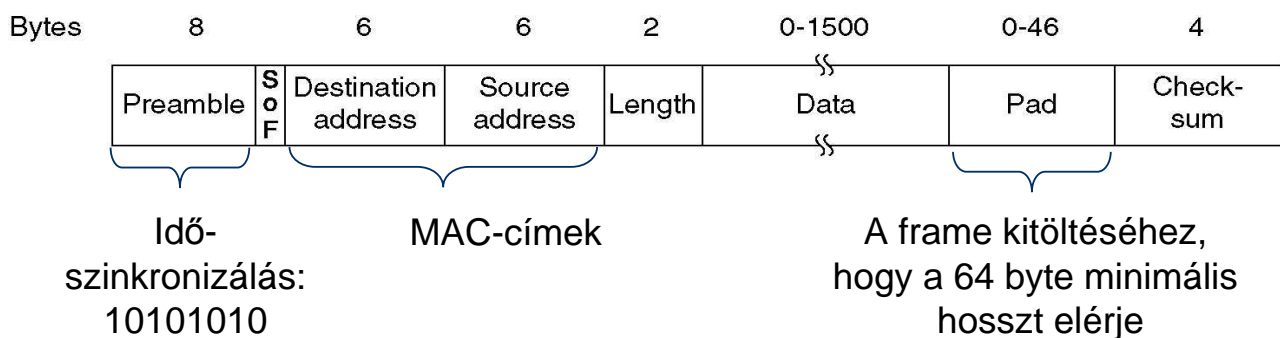


Az Ethernet fizikai rétege

- Médiemtől függő
- Tipikusan: Manchester kód
 - +/- 0.85 V
- Kód megsértés jelzi a frame-ek határát

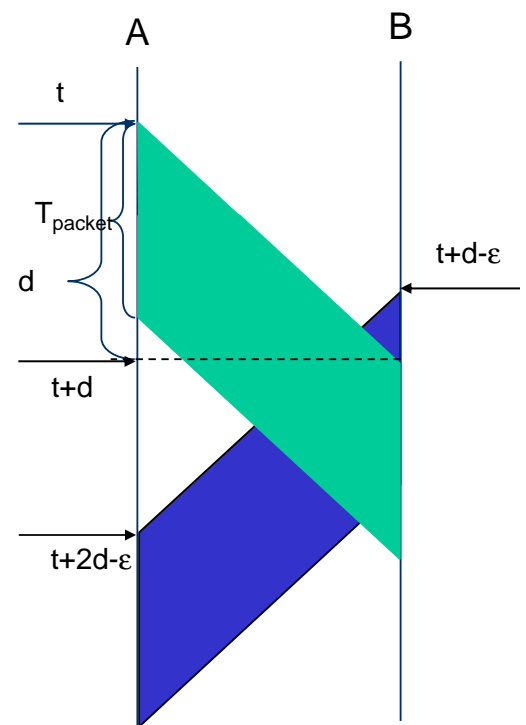
Az Ethernet adatkapcsolati rétege, MAC alrétege

- Az állomások, melyek egy kábelhez csatlakoznak egy **ütközési tartományt (collision domain)** definiálnak.
 - minden kapcsolódó állomás hall mindent
- MAC: lényegében CSMA/CD, binary exponential backoff
- Frame formátum:



Ethernet: Collision Detection -- Minimum Packet Size

- Ethernet minimum packet size = 64 byte = 512 bit
- Miért?
- Emlékezzünk, mi történik, ha két állomás A és B nagyon rövid frame-eket küldene
 - A küld egy csomagot
 - közvetlenül, mielőtt B észlelné ezt, B is elkezd küldeni
 - ez kollíziót okoz, amit B detektál
 - hogy A garantáltan detektálja ezt a kollíziót, az kell, hogy a csomag generálásához szükséges idő $T_{\text{packet}} \geq 2d$
- Ha A és B a kábel két legtávolabbi pontján van:
→ $T_{\text{min packet size}} \geq 2 \times \text{max propagation delay}$



Ethernet: End-to-End késés

- Miért 512 bit a minimális csomag méret?
- c kábelben = 60% * c vákuumban = 1.8×10^8 m/s
- 10Mbps Ethernet
 - A maximális konfigurált Ethernet hossza: 2,5km, ráta: 10Mbps
 - delay = $2500 \text{ m} / 1,8 \times 10^8 \text{ m/s} \approx 12.5\mu\text{s}$
 - +bevezetett repeaterok (max. 4 repeater: max. 5 szegmens)
 - legrosszabb esetben: $2 \times \text{max prop delay} = 51.2\mu\text{s}$
- $51.2\mu\text{s} \times 10\text{Mbps} = 512\text{bit}$ tehát a minimális csomag méret (512 bit van éppen „úton” a kábelben)
 - $51.2\mu\text{s}$ után a küldőnek garantált az egyedüli hozzáférés a linkhez
 - $51.2\mu\text{s}$: slot time az exponential backoff-ban

Ethernet: Csomagméret

- Mi a helyzet a maximális csomagmérettel?
 - Szükséges ahhoz, hogy egy csomópont ne sajátíthassa ki a hálózatot
 - 1500 byte az Ethernet-ben

Fast Ethernet

- Eredetileg az Ethernet 10 MBit/s átviteli rátát ért el
- Fast Ethernet
 - Cél: Hátrafele kompatibilitás
 - Eredmény: 802.3u Fast Ethernet (standard 1995)
- Fast Ethernet
 - Frame formátum, protokoll azonos maradt az eredetivel
 - A bitátviteli rátát 100 MBit/s-re növeli
 - Ennek következtében csökkenti a maximális kábelhosszt (és az egy szegmensen megengedett repeater-ek számát)

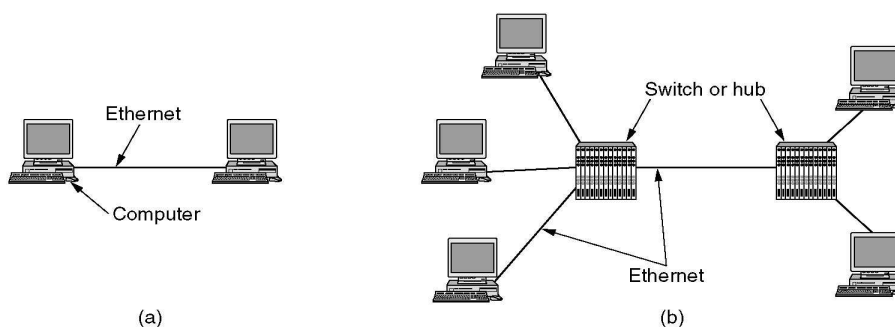
Fast Ethernet – Vezetékek

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

- Standard category-3 twisted pair (telefon kábel) nem támogat 200 MBaud rátát 100 m-en (100Mbps Manchester kóddal)
 - Megoldás: 2 kábelpár csökkentett rátával
- Manchester helyett 4B/5B-kód Cat-5-kábelen

Gigabit Ethernet

- Gigabit Ethernet
 - Cél: a korábbi Ethernet standard messzemenő átvétele
 - Erdmény: 802.3z Gigabit Ethernet (standard 1998)
- Ennek az ára: korlátozás pont-pont kapcsolatra,
 - Minden kábelhez pontosan két állomás kapcsolódik
 - vagy switch vagy hub



Gigabit Ethernet

- Switch esetén
 - Nincs kollízió → CSMA/CD nem szükséges
 - Full-duplex operációt tesz lehetővé minden linken
- Hub esetén
 - Kollíziók, fél-duplex operáció (azaz váltakozva simplex), CSMA/CD
 - Max. kábelhossz 25 m
- Carrier Extension:
 - Az Ethernet kompatibilitás megtartása miatt a „minimum packet size” nem változott. Ehelyett a küldő hardware az 512 byte-nál rövidebb frame-eket saját kitöltő jeleivel kiegészíti 512 byte hosszúra (padding). Ezt a fogadó hardware eltávolítja. Ennek a módszernek a neve „Carrier Extension”.
- Frame bursting:
 - Több rövid frame-et „egybefűzve” vihet át. Az összhosszt kitölti 512 byte-ra

Gigabit Ethernet – Vezetékek

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

LAN-ok összekapcsolása

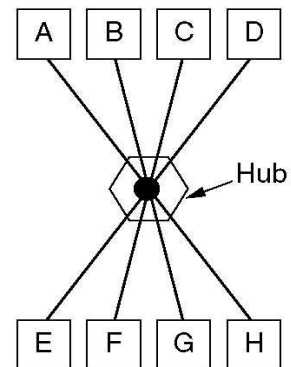
Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

Repeater

- Szignál-regenerátor
 - Fizikai réteg komponense
 - Két kábelt köt össze
 - Fogad egy szignált és azt regenerálva továbbítja a másik kábelben
 - Csak az elektromos vagy az optikai szignált továbbítja
 - A tartalmat (biteket) nem interpretálja
- Repeaterk a hálózatot fizikai szegmensekre osztják
 - A logikai topológia megmarad
 - A csatlakozó kábelek közös ütközési tartományt alkotnak

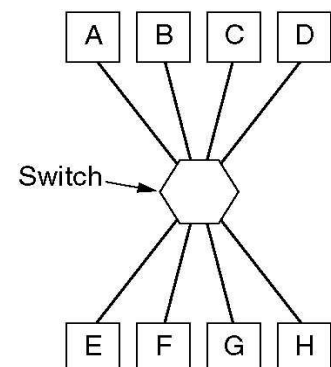
Hub

- Kábeleket köt össze csillag topológiában
 - Hasonló a Repeaterhez
 - A szignálokat minden csatlakozó kábelén továbbítja
 - Fizikai réteg komponense
 - A tartalmat nem interpretálja
 - A csatlakozó kábelek egy ütközési tartományt alkotnak



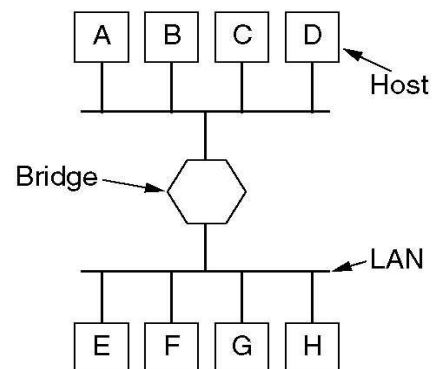
Switch

- Terminálokat csillag topológiába kapcsol össze
 - Adatkapcsolati réteg komponense
 - Kollíziók egy szegmensen belül maradnak
 - A frame-ek célcímét megvizsgálja és a frame-et csak a megfelelő kábelén továbbítja
 - ehhez szükséges puffer és
 - tudni kell melyik állomás hol csatlakozik
 - Egy táblázatot tart nyilván:
 - Megfigyeli, hogy honnan jön egy csomag, a küldőt azon a kábelén lehet elérni
 - Backward learning



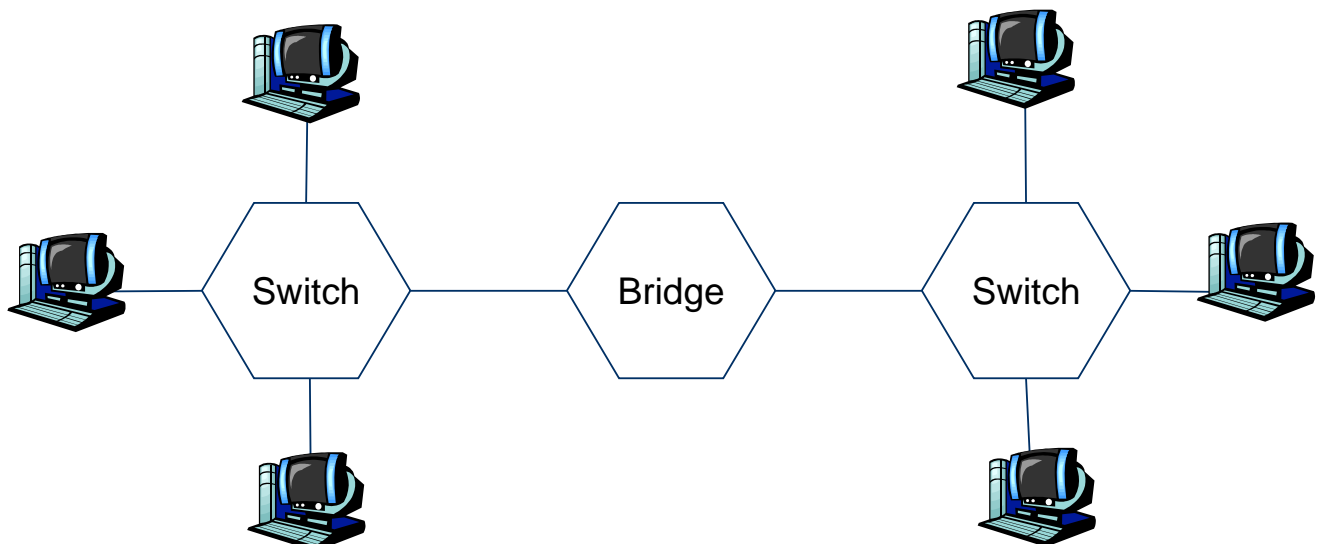
Bridge

- Lokális hálózatokat kapcsol össze
 - Ellentétben switch-ekkel (azok csak állomásokat -- eredetileg)
 - Adatkapcsolati réteg komponense
 - Elkülöníti a kollíziókat
 - Megvizsgálja az érkező frame-eket
 - A frame-et csak a megfelelő kábelen továbbítja
 - Csak korrekt frame-eket továbbít
 - Az átmenet bridge és switch között folyamatos
 - Összekapcsolhat többféle LAN típust



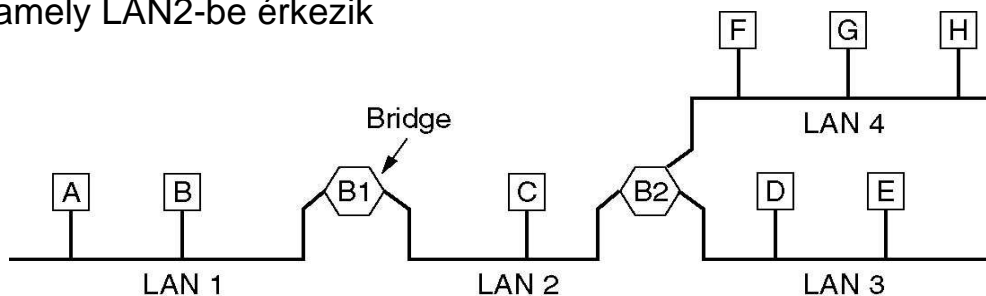
Switches & bridges

- Tipikus kombináció: bridge csak egy „másik állomás” a switch számára



Backward learning a bridge-ekben

- Backward learning trivialis switch-ekben – mi a helyzet a bridge-ekben?
- Példa: A küld frame-et E-nek
 - Tegyük fel, B1 és B2 tudja, hogy hol van E
 - B2 azt fogja látni, hogy A frame-je LAN2-ből jön
 - Mivel B2 nem tud LAN1-ről, B2 azt feltételezi, hogy A LAN2-ben van
 - Ami jó!
B1 továbbítani fog minden A-nak küldött csomagot LAN1-nek, amely LAN2-be érkezik

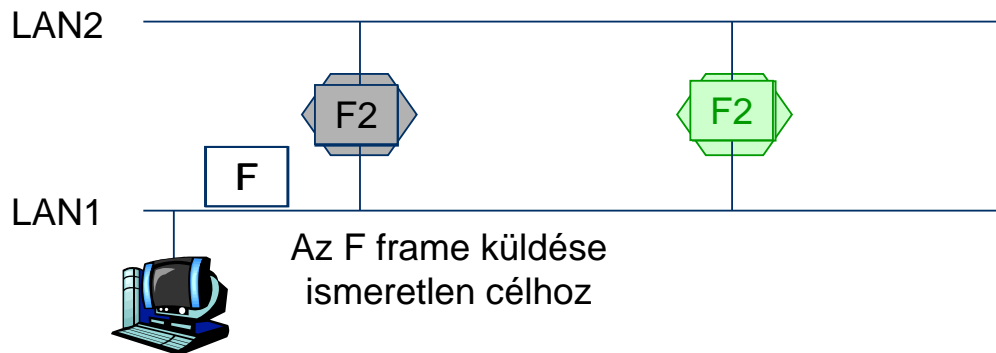


Backward learning a bridge-ekben – bootstrapping

- Az előző példában: honnan tudja B2 kezdetben, hogy hol van E?
- Válasz: NEM tudja
 - Opció 1: kézi konfiguráció – nem éppen szép megoldás!
 - Opció 2: nem számít – egyszerűen továbbítja az ismeretlen című csomagot mindenfele
 - Azon hálózat kivételével, ahonnan érkezett
- Az algoritmus:
 - elárasztás (flood) ha a cím ismeretlen;
 - dobja el ha tudja, hogy nem szükséges;
 - továbbítja specifikusan, ha a cél címe ismert

Elárasztás bridge által – problémák

- “Backward learning by flooding” egyszerű, de problémás
- Példa:
 - Egy második bridge is összeköti a két LAN-t a nagyobb megbízhatóság miatt



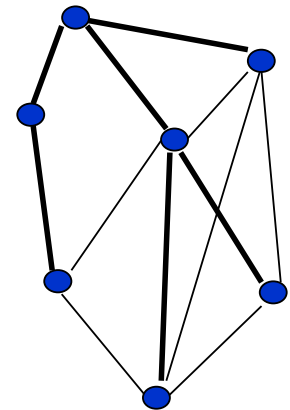
- F végtelen ciklusba kerül...
- Hogy kerüljünk el ilyen ciklusokat?

1. Megoldás: Valahogy korlátozzuk az elárasztást

- Korlátolatlan, brute-force flooding nyilvánvalóan rossz
 - Kerüljük el a ciklust azáltal, hogy **megjegyezzük**, hogy mely frame-ek azok, amelyeket már továbbítottunk
 - Ha már láttunk és továbbítottunk egy frame-et, dobjuk el
 - Előfeltétel: állapot és egyértelműség
 - Bridge-eknek meg kell jegyezni, hogy mely frame-eket továbbította
 - A frame-eknek egyértelműen azonosíthatóknak kell lenni – legalább küldő, fogadó és sorozatszám szükséges az azonosításhoz
- Nagy overhead!
- Különösen az állapotok tárolása a probléma, és a keresés a sok állapot között
 - Nem igen használják

2 Megoldás: Feszítőfák

- A csomagok ciklusai csak akkor jöhetnek létre, ha a gráf, amit a bridge-ek definiálnak kört tartalmaz
 - Tekintsük a LAN-okat és a bridge-eket csomópontoknak
 - Egy LAN-csomópont és egy bridge-csomópont össze van kötve egy éllel, ha a LAN a bridge-hez kapcsolódik
 - Redundáns élek köröket formálnak ebben a gráfban
- Ötlet: alakítsuk át a gráfot köröktől mentessé
- Legegyszerűbb megoldás: Számítsunk ki egy feszítőfát ebben a LAN-bridge gráfban
 - Definíció: Legyen $G=(V,E)$ egy gráf. G egy olyan $T=(V, E_T)$ részgráfját, $E_T \subseteq E$, ami egy fa (összefüggő és nem tartalmaz kört), G **feszítőfájának** nevezzük
 - Egyszerű, önkonfiguráló, nem kell kézi beavatkozás
 - De nem optimális: az installált bridge-ek kapacitását nem biztos hogy kihasználja
 - IEEE 802.1D: Spanning Tree Protocol (STP),
IEEE 802.1w: Rapid Spanning Tree Protocol (RSTP)



Egy feszítőfa

Konvergencia: Switch és bridge

- Tradícionálisan, a megkülönböztetés bridge és switch között értelmes volt
- Ma: a legtöbb készülék kínálja mindkét típusú funkcionalitást
- Gyakran inkább marketing megkülönböztetés, mint műszaki