

## Számítógépes Hálózatok és Internet Eszközök

2007

### 13. Adatkapcsolati réteg – LAN-ok összekapcsolása

## LAN-ok összekapcsolása

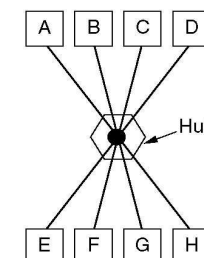
Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

## Repeater

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

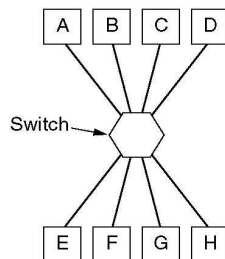
## Hub

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



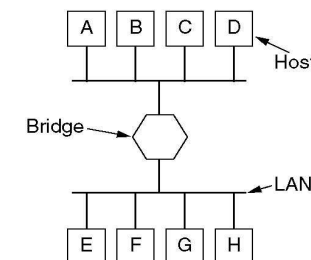
## Switch

- Terminálokat csillag topológiába kapcsol össze
  - Adatkapcsolati réteg komponense
  - Kollíziók egy szegmensen belül maradnak
  - A frame-ek cél címét megvizsgálja és a frame-et csak a megfelelő kábelben 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 kábelen lehet elérni
    - Backward learning



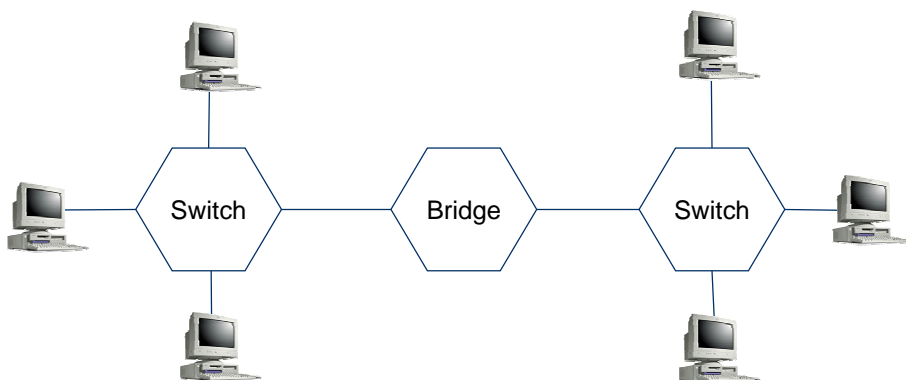
## Bridge

- 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 kábelben továbbítja
  - Csak korrekt frame-eket továbbít
  - Az átmenet bridge és switch között folyamatos
  - Összekapcsolhat többféle LAN típust



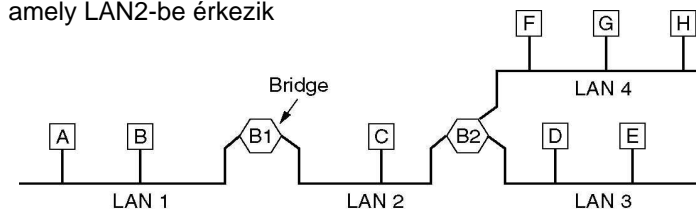
## Switches & bridges

- 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

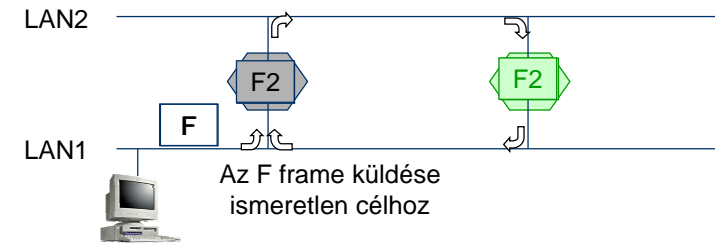


## Backward learning a bridge-ekben – bootstrapping

- 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, pl. a nagyobb megbízhatóság miatt



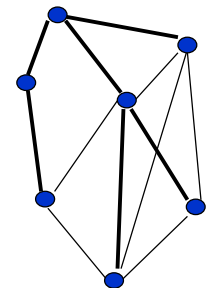
- F végtelen ciklusba kerül...
- Hogy kerülünk el ilyen ciklusokat?

## 1. Megoldás: Valahogy korlátozzuk az elárasztást

- Korlátozatlan, 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:
    - 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 köreket 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)



## Spanning Tree Protocol (STP) (IEEE 802.1D)

- Minden bridge-nek van egy azonosító száma, amely a MAC címen plusz egy konfigurálható prioritáson alapul
- Az a bridge lesz a feszítőfa gyökere, amelynek minimális az azonosítója
  - Először a prioritást hasonlítjuk össze az azonosítóban
  - Ha ez egyenlő, akkor a MAC cím dönt
  - Ezáltal a hálózat adminisztrátora tudja meghatározni a gyökér bridge-t
- Minden linknek van egy költsége
  - Konfigurálható az adminisztrátor által
  - Különböző technológiáknak különböző default költsége van

Pl.:

Sávszélesség	STP költség
10 Mbps	100
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

## Spanning Tree Protocol (IEEE 802.1D)

- Minden bridge meghatározza a legalacsonyabb költségű utat a gyökérhez
  - Azok a portok, amelyek ezen az úton vannak, un. *root portok* lesznek
- Az egy szegmensen lévő bridge-k közösen meghatározzák, hogy melyiküknek minimális a költsége a gyökérhez.
  - A port, amelyen a szegmens ehhez a bridge-hez kapcsolódik, *kitüntetett port* lesz
- Minden port blokkolódik, amely nem root port és nem kitüntetett port

## Spanning Tree Protocol (IEEE 802.1D)

- Ha a legkisebb költségű út nem egyértelmű:
  - Ha egy bridge-től több minimális költségű út van a gyökérhez, azt az utat választjuk ezek közül, amelyen a következő bridge azonosítója minimális
  - Ha egy szegmenstől több bridge-en keresztül vezet minimális költségű út a gyökérhez, azt az utat választjuk ezek közül, amelyen a következő bridge azonosítója minimális
  - Ha két bridge több kábellel van összekötve és egy bridge-en több port is root port lehetne, válasszuk a legalacsonyabb számú portot root portnak

## STP Algorithm

- Az STP algoritmusban az üzenetekhez a bridge-k speciális frame-eket, un. Bridge Protocol Data Unit (BPDU) használnak.
- A gyökér meghatározása:
  - A bridgek meghirdetik az azonosítókat.  
Ha változik a legalacsonyabb azonosító, amit hallottak, továbbítják a szomszédoknak.
  - A legalacsonyabb azonosító eljut minden bridge-hez:  
A legalacsonyabb azonosítójú bridge lesz a gyökér.
- Ezután a költségek a gyökértől elárasztással terjednek a hálózaton.
  - Minden bridge figyeli a legalacsonyabb költségű utat a gyökérhez.
  - Ha ez a költség változik, a bridge továbbítja ezt a szomszédai felé.
  - Nemegetertelműség esetén a bridge-azonosítók alapján dönt.
- A gyökér szabályos intervallumokban "Hello" broadcast-üzenetet küld.



## Konvergencia: Switch és bridge

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