Számítógépes Hálózatok

5. Előadás: Adatkapcsolati réteg

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

Adatkapcsolati réteg

- Szolgáltatás
  - Adatok keretekre tördelése: határok a csomagok között
  - Közeghozzáférés vezérlés (MAC)
  - Per-hop megbízhatóság és folyamvezérlés
- Interfész
  - Keret küldése két közös médiumra kötött eszköz között
- Protokoll
  - Fizikai címzés (pl. MAC address, IB address)
- Példák: Ethernet, Wifi, InfiniBand

Adatkapcsolati réteg

- Funkciók:
  - Adat blokkok (keretek/frames) küldése eszközök között
  - A fizikai közeghez való hozzáférés szabályozása
- Legfőbb kihívások:
  - Hogyan keretezzük az adatokat?
  - Hogyan ismerjük fel a hibát?
  - Hogyan vezéreljük a közeghozzáférést (MAC)?
  - Hogyan oldjuk fel vagy előzzük meg az ütközési helyzeteket?

Forgalomszabályozás

Forgalomszabályozás

- gyors adó lassú vevő problémája (*elárasztás*)
- még hibamentes átvitel esetén se lesz képes a vevő kezelni a bejövő kereteket

Megoldási lehetőségek

1. visszacsatolás alapú forgalomszabályozás (avagy angolul *feedback-based flow control*)
   - engedélyezés
2. Sebesség alapú forgalomszabályozás (avagy angolul *rate-based flow control*)
   - protokollba integrált sebességkorlát
   - az adatkapcsolati réteg nem használja

Elemi adatkapcsolati protokollok

Feltevések

- A fizikai, az adatkapcsolati és a hálózati réteg független folyamatok, amelyek üzeneteken keresztül kommunikálnak egymással.
- Az *A* gép megbízható, összeköttetés alapú szolgálat alkalmazásával akar a *B* gépnek egy hosszú adatfolyamot küldeni. (Adatok előállítására sosem kell várnia A gépnek.)
- A gépek nem fagynak le.
- Adatkapcsolati fejrészben vezérlési információk; adatkapcsolati lábrészben ellenőrző összeg

Kommunikációs fajták

- *szimplex kommunikáció* – a kommunikáció pusztán egy irányba lehetséges
- *fél-duplex kommunikáció* – mindkét irányba folyhat kommunikáció, de egyszerre csak egy irány lehet aktív.
- *duplex kommunikáció* – mindkét irányba folyhat kommunikáció szimultán módon

Korlátozás nélküli szimplex protokoll

a legegyszerűbb protokoll („utópia”)

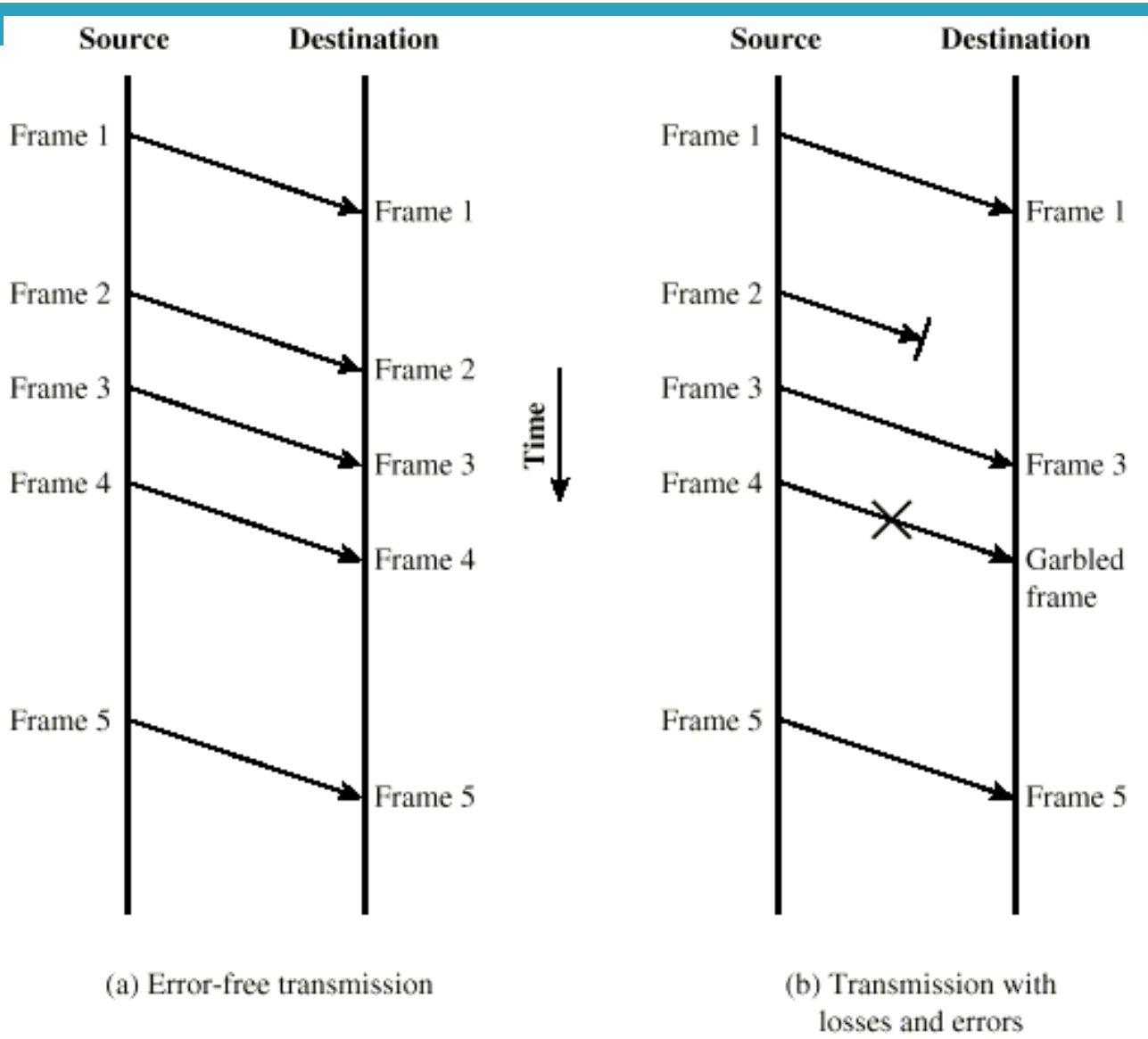
A környezet

- mind az adó, mind a vevő hálózati rétegei mindig készen állnak;
- a feldolgozási időktől eltekintünk;
- végtelen puffer-területet feltételezünk;
- Az adatkapcsolati rétegek közötti kommunikációs csatorna sosem rontja vagy veszíti el a kereteket;

A protokoll

- résztvevők: *küldő* és *vevő;*
- nincs sem sorszámozás, sem nyugta;
- küldő végtelen ciklusban küldi kifele a kereteket folyamatosan;
- a vevő kezdetben várakozik az első keret megérkezésére, keret érkezésekor a hardver puffer tartalmát változóba teszi és az adatrészt továbbküldi a hálózati rétegnek

Átvitel hiba nélkül és hibával



(a) Error-free transmission

(b) Transmission with losses and errors

Szimplex megáll-és-vár protokoll (stop-and-wait protocol)

A környezet

- mind az adó, mind a vevő hálózati rétegei mindig készen állnak;
- A vevőnek Δt időre van szüksége a bejövő keret feldolgozására (nincs pufferelés és sorban állás sem);
- Az adatkapcsolati rétegek közötti kommunikációs csatorna sosem rontja vagy veszíti el a kereteket;

A protokoll

- résztvevők: *küldő* és *vevő;*
- küldő egyesével küldi kereteket és addig nem küld újat, még nem kap nyugtát a vevőtől;
- a vevő kezdetben várakozik az első keret megérkezésére, keret érkezésekor a hardver puffer tartalmát változóba teszi és az adatrészt továbbküldi a hálózati rétegnek, végül nyugtázza a keretet

Következmény: fél-duplex csatorna kell.

Szimplex protokoll zajos csatornához

A környezet

- mind az adó, mind a vevő hálózati rétegei mindig készen állnak;
- A vevőnek Δt időre van szüksége a bejövő keret feldolgozására (nincs pufferelés és sorban állás sem);
- Az adatkapcsolati rétegek közötti kommunikációs csatorna hibázhat (keret megsérülése vagy elvesztése);

A protokoll

- résztvevők: *küldő* és *vevő;*
- küldő egyesével küldi kereteket és addig nem küld újat, még nem kap nyugtát a vevőtől egy megadott határidőn belül, ha a határidő lejár, akkor ismételten elküldi az aktuális keretet;
- a vevő kezdetben várakozik az első keret megérkezésére, keret érkezésekor a hardver puffer tartalmát változóba teszi, leellenőrzi a kontroll összeget,
  - ha nincs hiba, az adatrészt továbbküldi a hálózati rétegnek, végül nyugtázza a keretet;
  - Ha hiba van, akkor eldobja a keretet és nem nyugtáz.

Következmény: duplikátumok lehetnek.

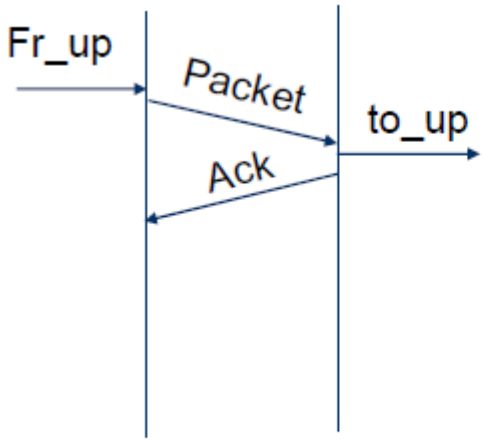
Megáll-és-vár

Egyszerű de nem hatékony nagy távolságok és nagy sebességű hálózat esetén.

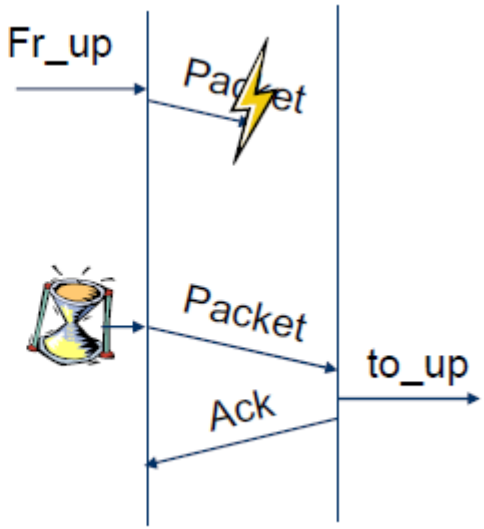
Küldhetnénk egymás után folyamatosan???

Mi is a probléma?

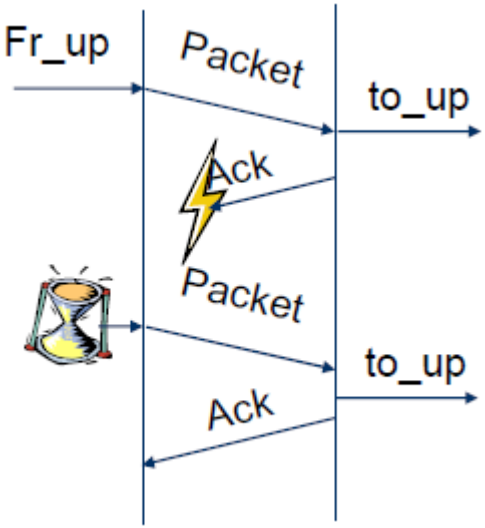
Általában



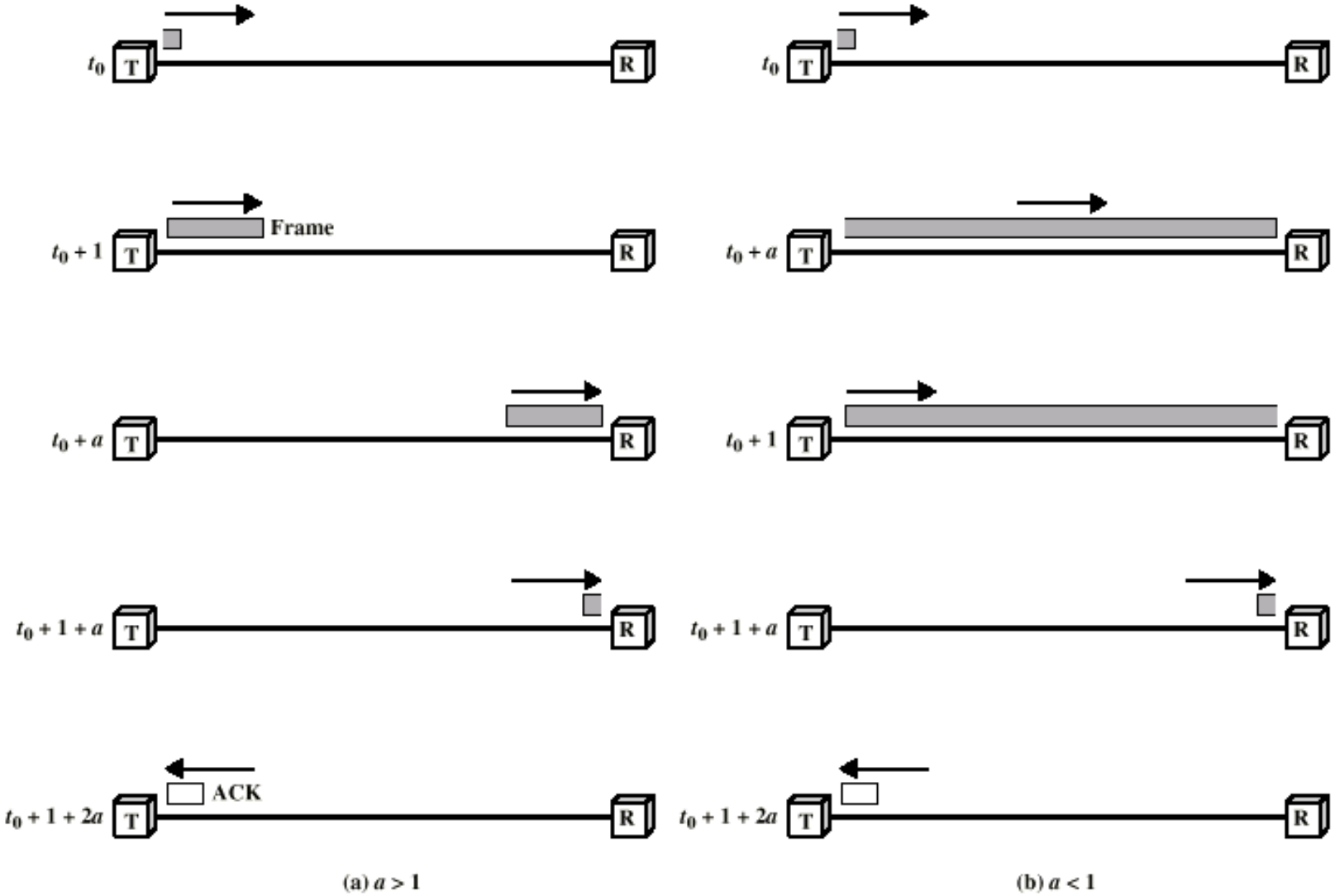
Csomagvesztés esetén



ACK vesztés esetén



Csatorna kihasználtság



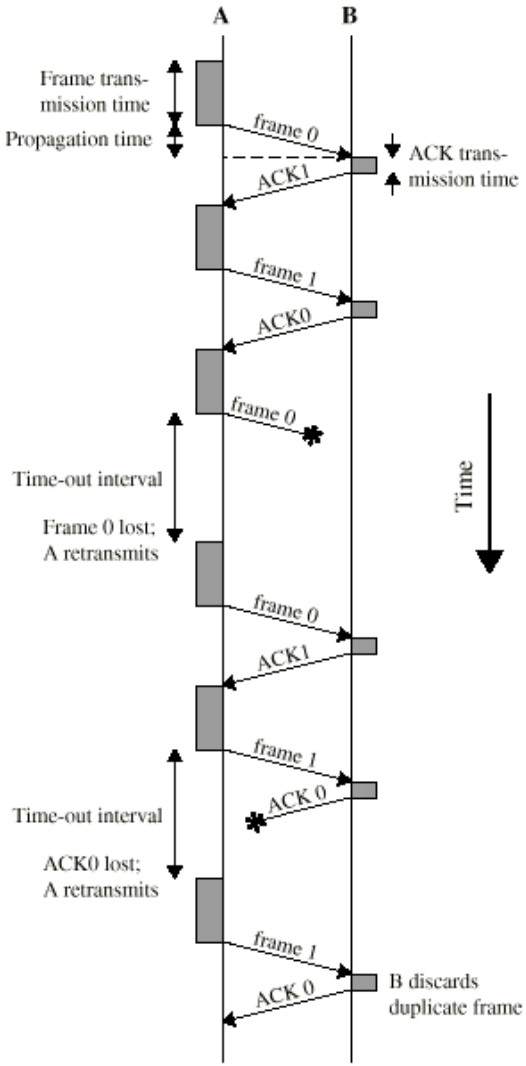
Alternáló-bit protokoll (ABP)

- **Megoldás:** sorszámok használata
- Mennyi sorszámra lesz szükség? $\{0,1\}$ elegendő

A protokoll (*ARQ*) – Alternáló-bit protokoll

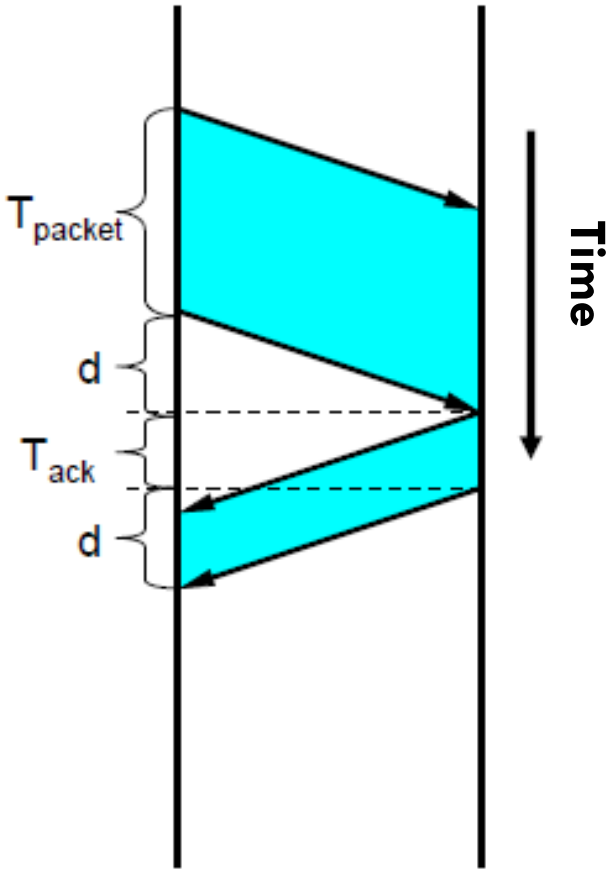
- résztvevők: *küldő* és *vevő;*
- küldő egyesével küldi a sorszámmal ellátott kereteket (kezdetben 0-s sorszámmal) és addig nem küld újat, még nem kap nyugtát a vevőtől egy megadott határidőn belül:
  - ha a nyugta megérkezik a határidőn belül, akkor lépteti a sorszámot *mod 2* és küldi a következő sorszámmal ellátott keretet;
  - ha a határidő lejár, akkor ismételten elküldi az aktuális sorsszámmal ellátott keretet;
- a vevő kezdetben várakozik az első keret megérkezésére 0-s sorszámmal, keret érkezésekor a hardver puffer tartalmát változóba teszi, leellenőrzi a kontroll összeget és a sorszámot
  - ha nincs hiba, az adatrészt továbbküldi a hálózati rétegnek, végül nyugtázza a keretet és lépteti a sorszámát *mod 2*;
  - ha hiba van, akkor eldobja a keretet és nem nyugtáz.

ABP


A
B
Frame trans-
mission time
Propagation time
frame 0
ACK trans-
mission time
ACK1
frame 1
ACK0
frame 0
Time-out interval
Frame 0 lost;
A retransmits
Time
frame 0
ACK1
frame 1
ACK 0
Time-out interval
ACK0 lost;
A retransmits
frame 1
ACK 0
B discards
duplicate frame

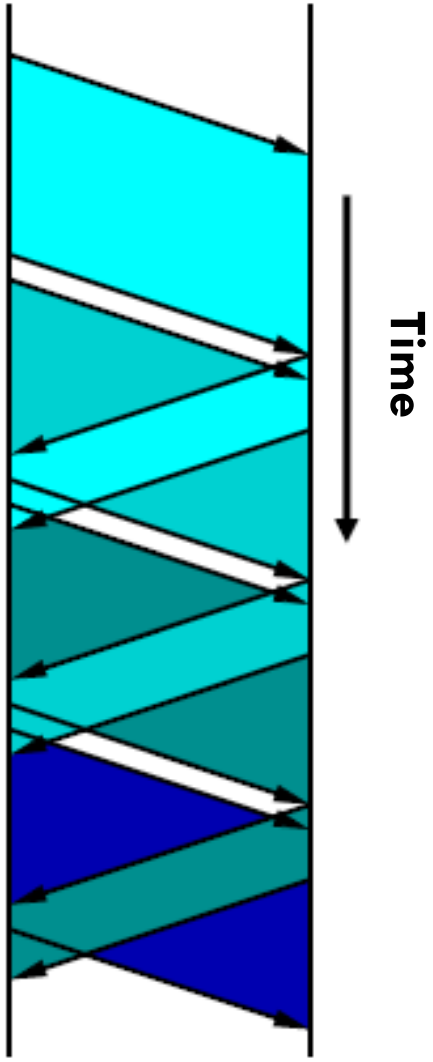
ABP – Csatorna kihasználtság

- Kihasználtság (η) a következő két elem aránya
  - A csomag elküldéséhez szükséges idő (T_{packet})
  - Az idő, ami a következő keret küldéséig eltelik
    - Az ábrán: (T_{packet} + d + T_{ack} + d)
- ABP esetén:
  - $\eta = T_{packet} / (T_{packet} + d + T_{ack} + d)$
- Nagy propagációs idő esetén az ABP nem hatékony



Hogyan javítsunk a hatékonyságon?

- A küldők egymás után küldik a kereteket
  - Több keretet is kiküldünk, nyugta megvárása nélkül.
  - Pipeline technika
- ABP kiterjesztése
  - Sorszámok bevezetésével



Csúszó-ablak protokollok 1/2

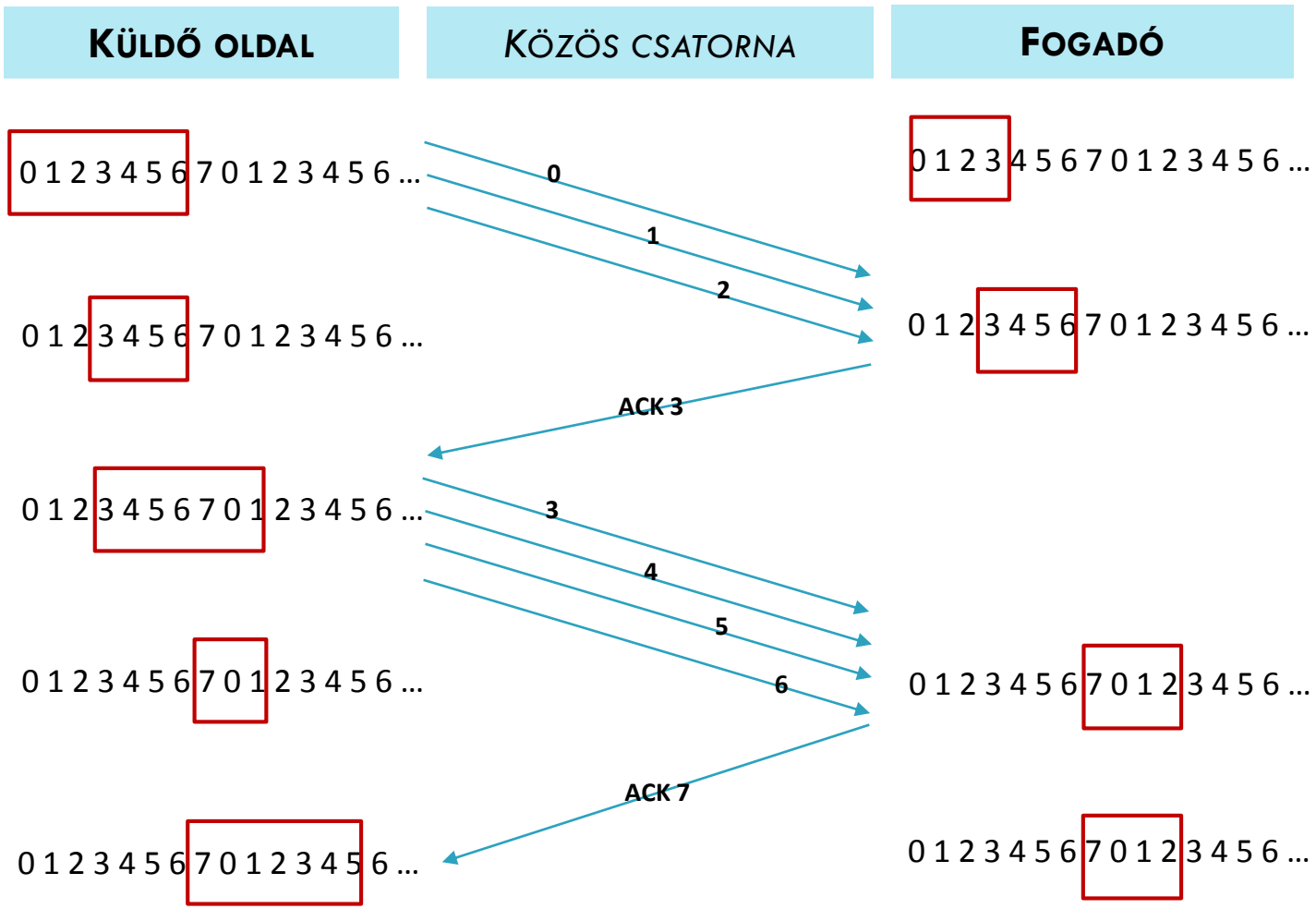
ALAPOK (ÁLTALÁNOS)

- 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.

Példa 3-bites csúszó-ablak protokollra



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.

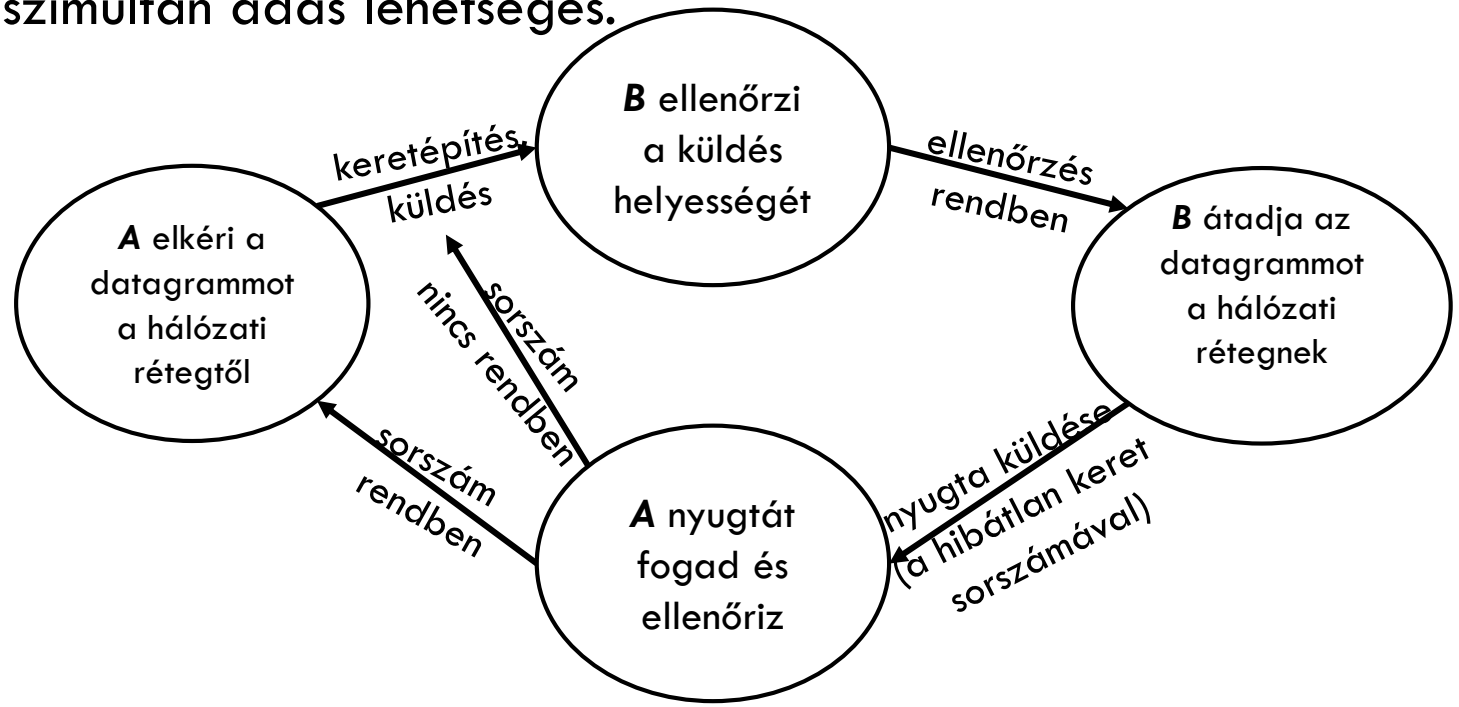
JELLEMZŐK (GYAKORLATI ALKALMAZÁS ESETÉN)

- 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);

Egybites csúszó-ablak protokoll állapotátmenetei

KÖRNYEZET

- A maximális ablak méret legyen *1*.
- *Emlékeztetőül*: két irányú adatforgalom lehetséges, azaz szimultán adás lehetséges.



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*

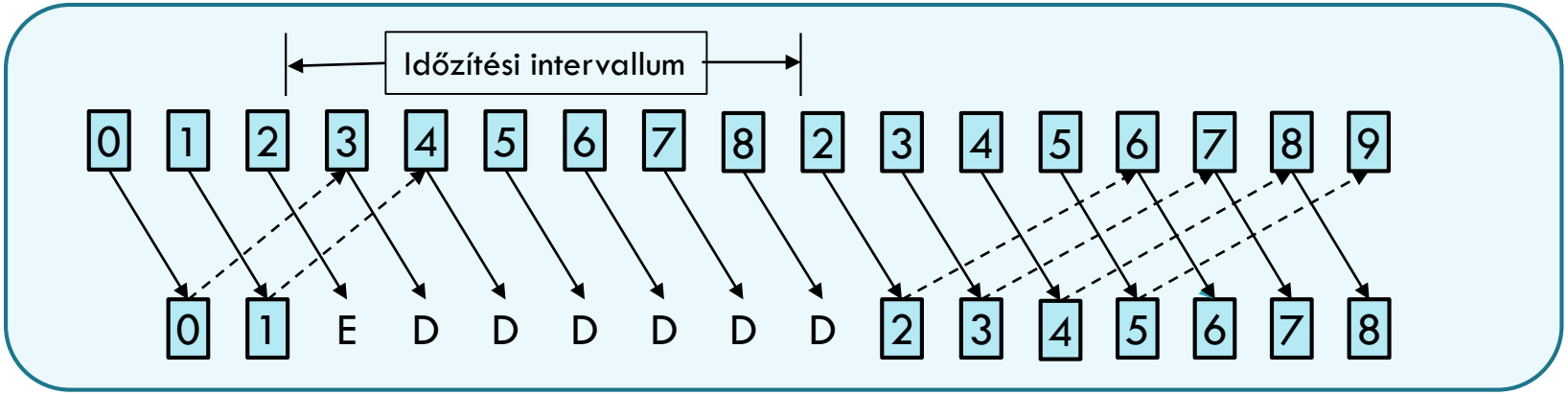
„visszalépés N-nel” stratégia

Stratégia lényege

- Az összes hibás keret utáni keretet eldobja és nyugtát sem küld róluk.
- Mikor az adónak lejár az időzítője, akkor újraküldi az összes nyugtázatlan keretet, kezdve a sérült vagy elveszett kerettel.

Következmények

- Egy méretű vételi ablakot feltételezünk.
- Nagy sávszélességet pazarolhat el, ha nagy a hibaarány.



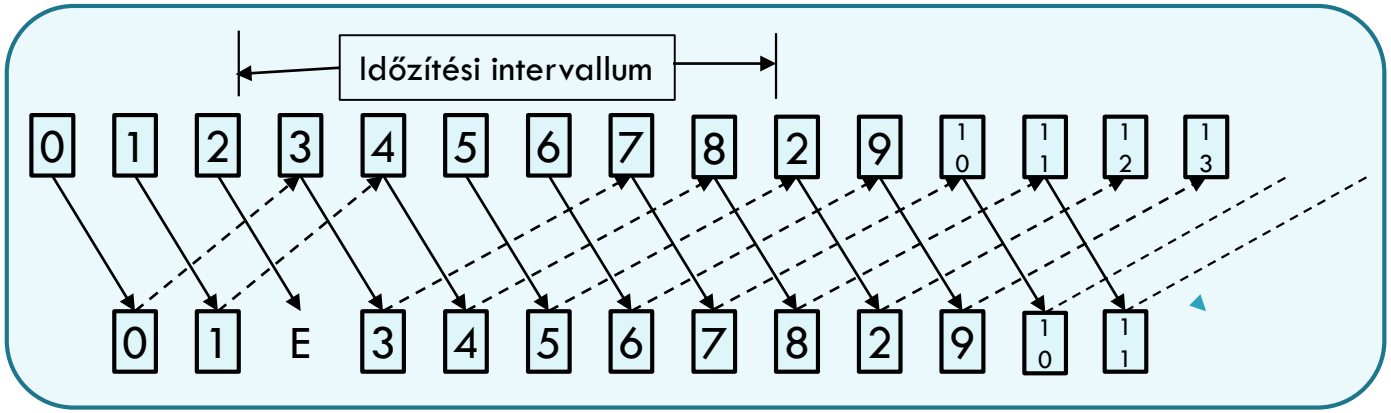
„ szelektív ismétlés” stratégia

Stratégia lényege

- A hibás kereteket eldobja, de a jó kereteket a hibás után puffereli.
- Mikor az adónak lejár az időzítője, akkor a legrégebbi nyugtázatlan keretet küldi el újra.

Következmények

- Javíthat a hatékonyságon a negatív nyugta használata. (*NAK*)
- Egynél nagyobb méretű vételi ablakot feltételezünk.
- Nagy memória igény, ha nagy vételi ablak esetén.



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[note 2]–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 ?

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

MAC alréteg

- 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

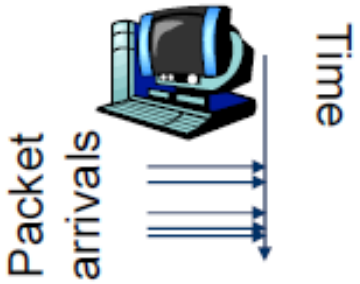
Frekvenciaosztásos nyalábolás

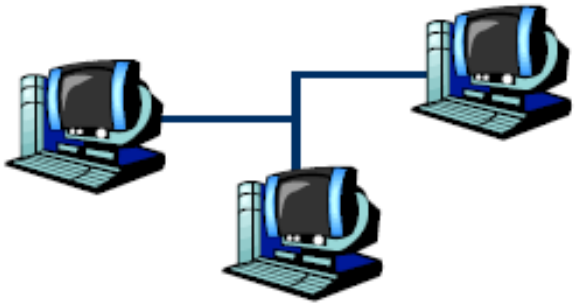
- *N* darab felhasználót feltételezünk, a sávszélet *N* egyenlő méretű sávra osztják, és minden egyes sávhoz hozzárendelnek egy felhasználót.
- Következésképpen az állomások nem fogják egymást zavarni.
- Előnyös a használata, ha fix számú felhasználó van és a felhasználók nagy forgalmi igényt támasztanak.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

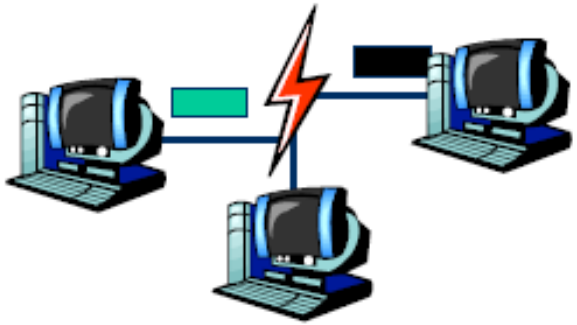
Időosztásos nyalábolás

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

Dinamikus csatornakiosztás







1. Állomás modell
   - N terminál/állomás
   - Annak a valószínűsége, hogy Δt idő alatt csomag érkezik λΔt, ahol λ az érkezési folyam rátája.

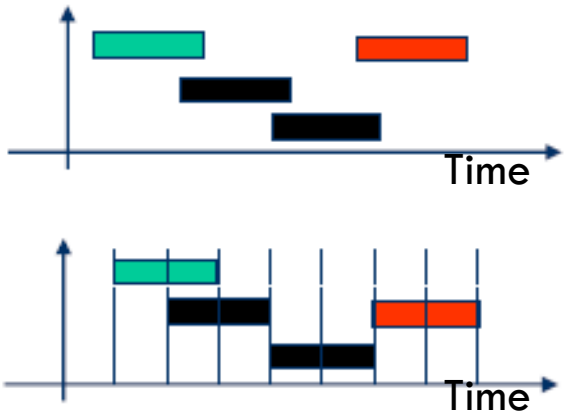
2. Egyetlen csatorna feltételezés
   - Minden állomás egyenrangú.
   - Minden kommunikáció egyazon csatornán zajlik.
   - Minden állomás tud ezen küldeni és fogadni csomagot.

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

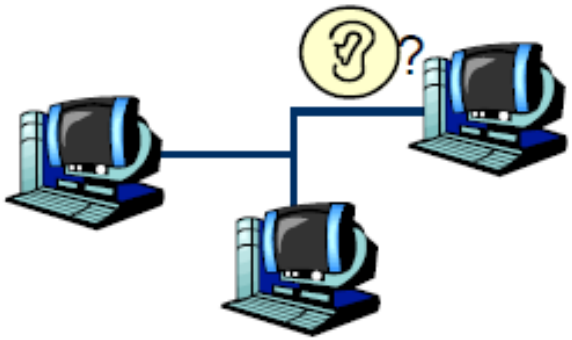


Használt időmodell

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 elkezdenek 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.

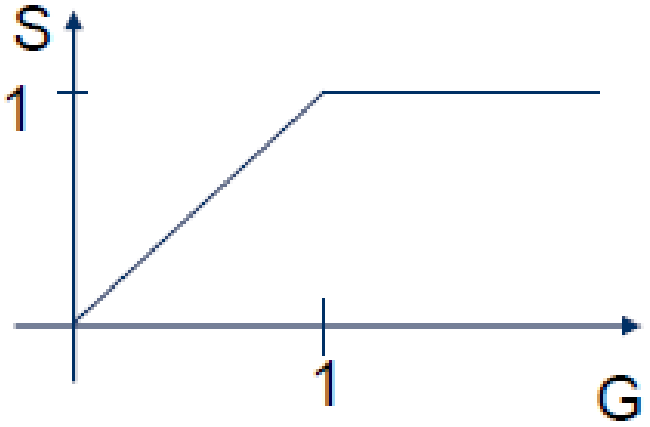
Megjegyzés: Ez egy egyszerűsített modell!

Hogyan mérjük a hatékonyságot?

- **Átvitel [Throughput] (S)**
  - A sikeresen átvitt csomagok/keretek száma egy időegység alatt
- **Késleltetés [Delay]**
  - Egy csomag átviteléhez szükséges idő
- **Fairség [Fairness]**
  - Minden állomás egyenrangúként van kezelve

Átvitel és terhelés

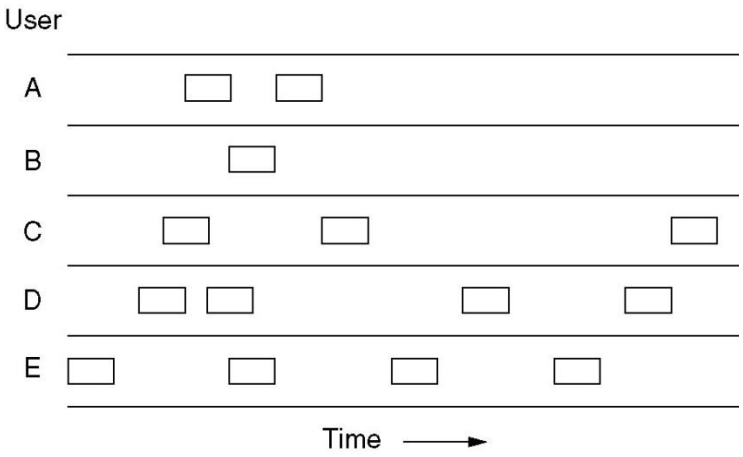
- **Terhelés (G)**
  - A protokoll által kezelendő csomagok száma egy időegység alatt (beérkező kérések)
  - G>1: túlterhelés
  - A csatorna egy kérést tud elvezetni

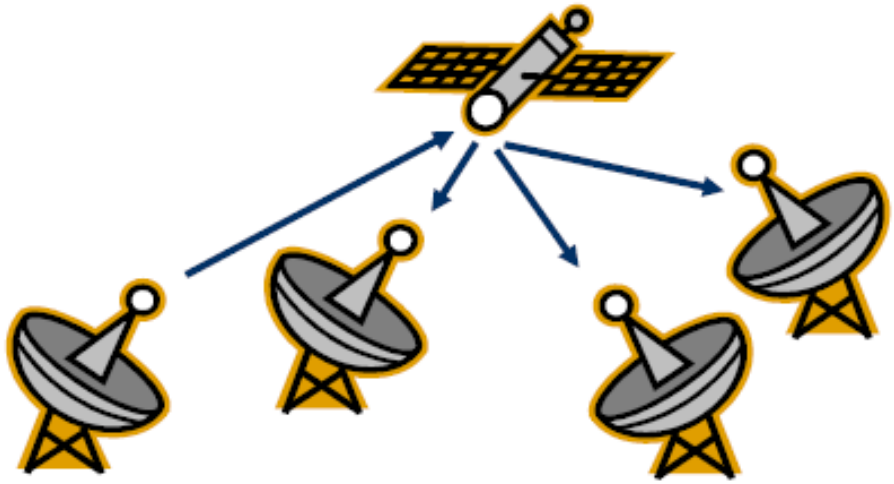


- **Ideális esetben**
  - Ha G<1, S=G
  - Ha G≥1, S=1
  - Ahol egy csomag kiküldése egy időegységet vesz igénybe.

(Tiszta) ALOHA

- Az algoritmust a 70-es években a Uni. of Hawaii fejlesztette
  - **Ha van elküldendő adat, akkor elküldi**
  - Alacsony költségű, nagyon egyszerű megoldás





ALOHA

- 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.
- Kévés küldő esetére készült

Teljesítmény elemzés -Poisson Folyam

- A „véletlen érkezések” egyik ünnepelt modellje a sorban-állás elméletben a Poisson folyam.
- A modell feltételezései:
  - Egy érkezés valószínűsége egy rövid Δt 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 Δt intervallum alatt közelít a nullához.

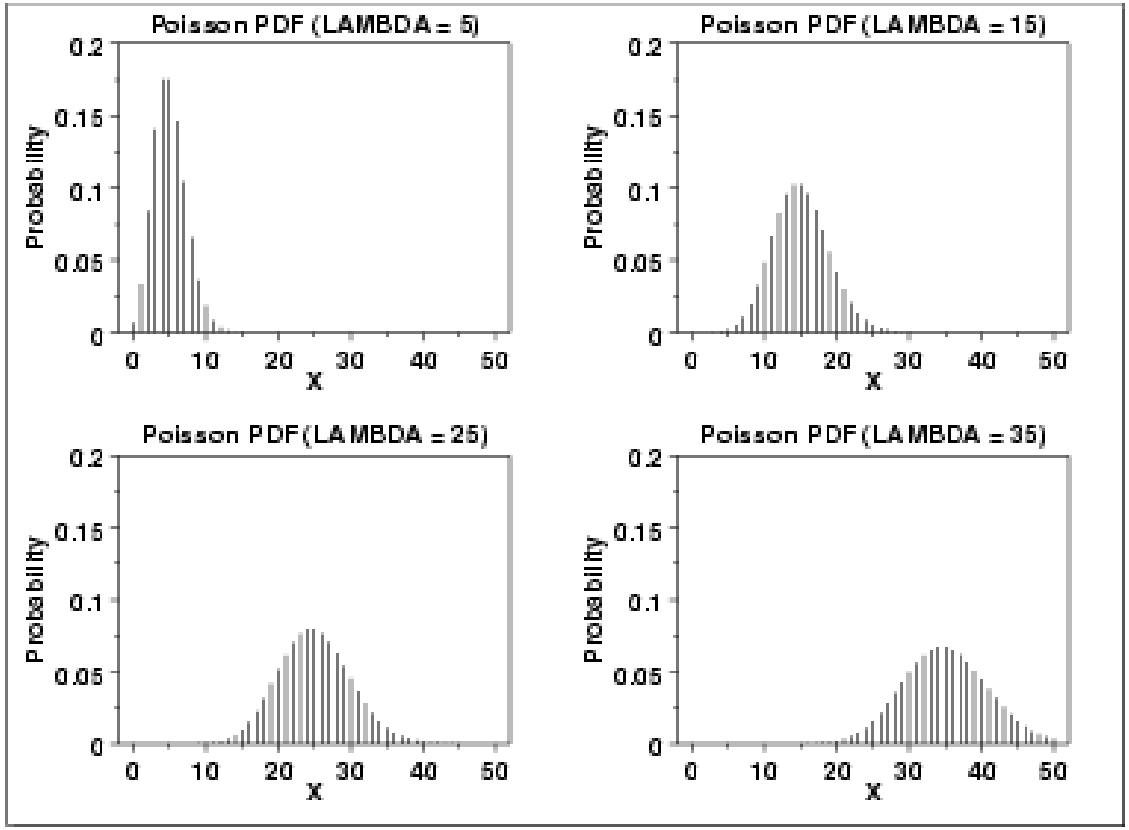
Teljesítmény elemzés –Poisson eloszlás

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

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

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

Poisson Eloszlás példák



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

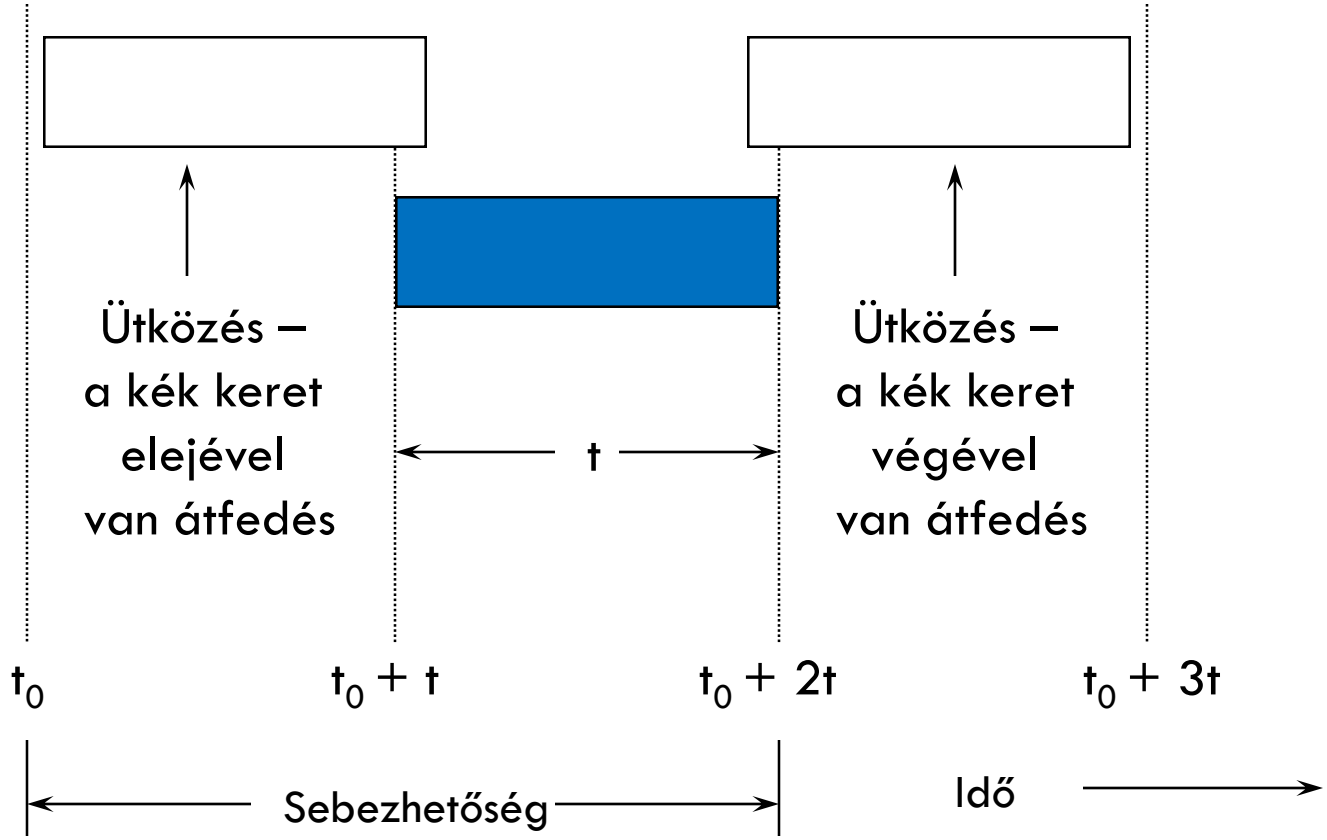
- Mivel S jelöli a „jó” átviteleket egy keret idő alatt és G jelöli az összes átviteli kísérletet egy keret idő alatt, így a következő összefüggést írhatjuk:

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

- 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



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)

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

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

ALOHA vizsgálata

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

$$G = 0.5,$$

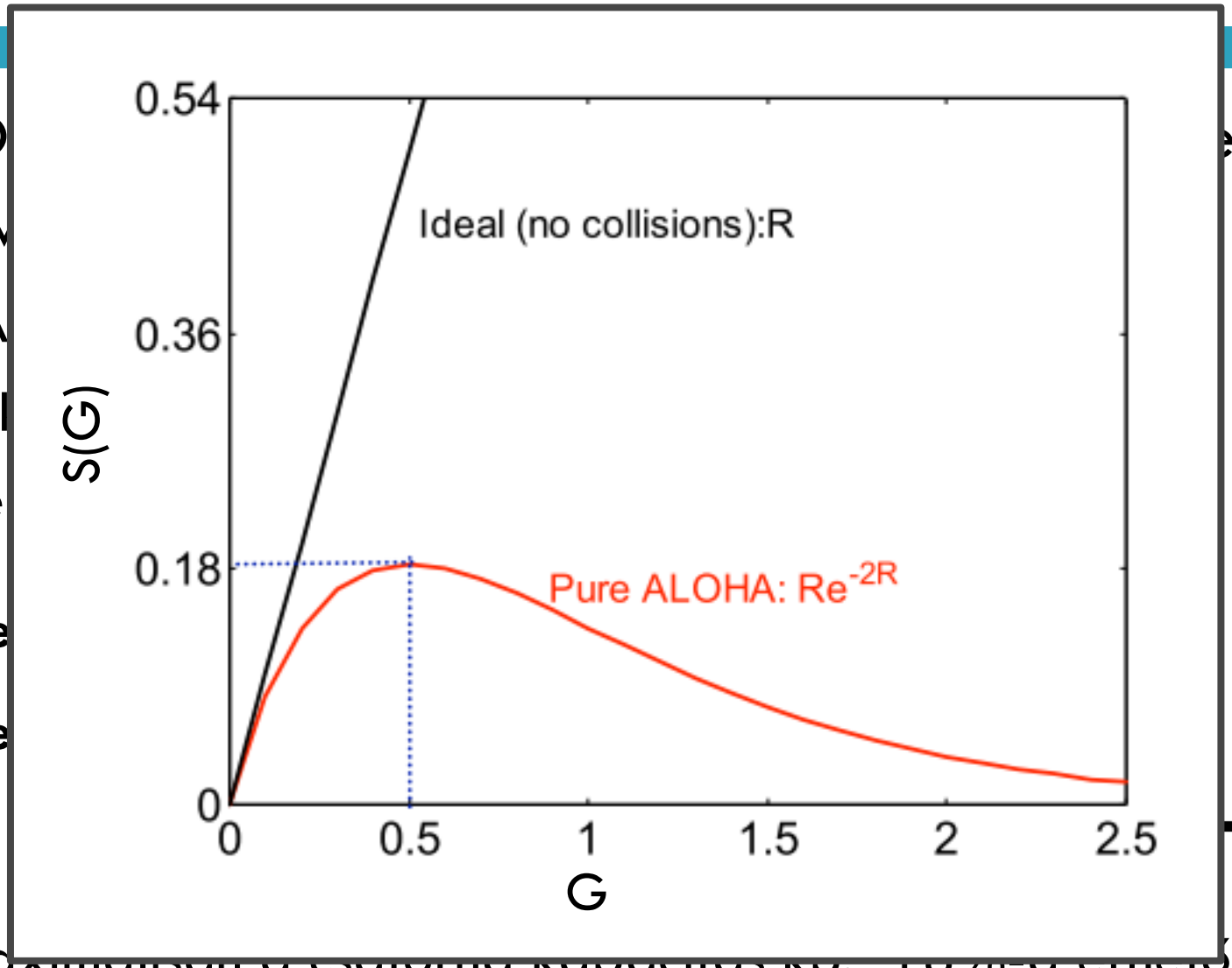
melyre $S(G) = 1/2e = 0.18$. Azaz a maximális throughput csak 18%-a a teljes kapacitásnak!!!

ALOHA vs TDMA

- A TD
  - A v
- Az A
  - Sol
  - De

Sende

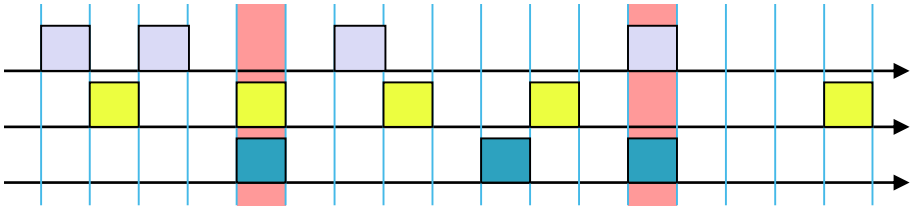
Sende



- Ma el

Réselt ALOHA

- 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:

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

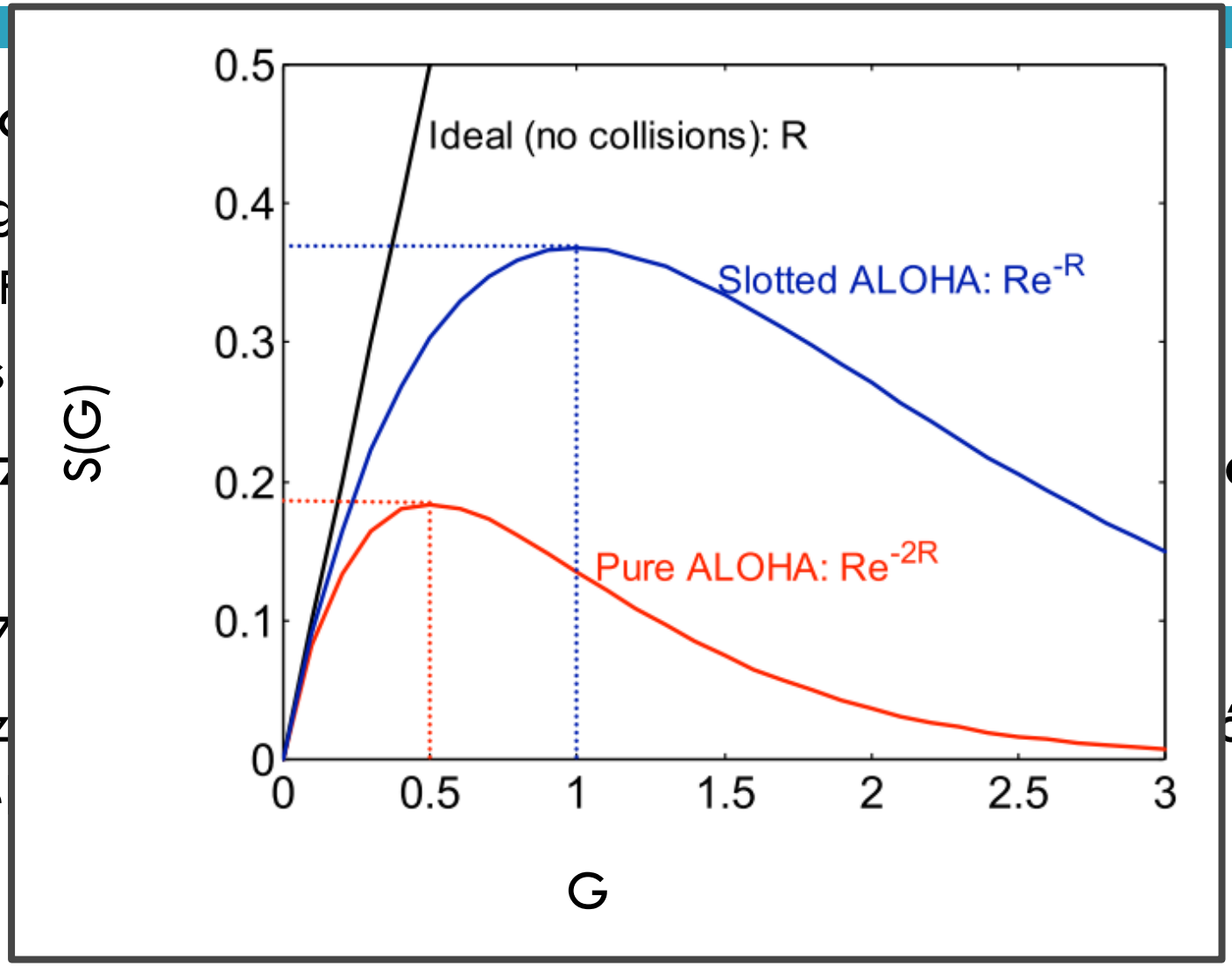
Ez esetben $t = T_f$ és továbbra is $k = 0$, amiből kapjuk, hogy:

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

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

Réselt ALOHA

- Proto
  - Ug
    - F
  - Cs
- Azaz általán nem
  - 37
  - Az órával ke



Köszönöm a figyelmet!