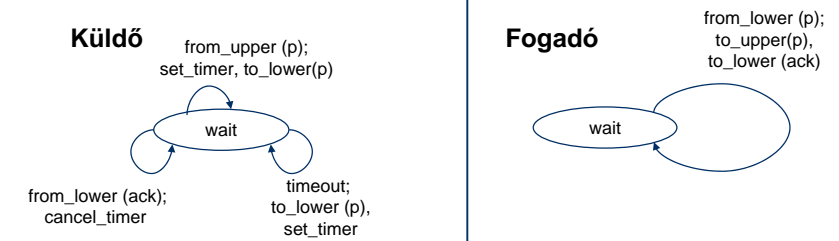


Számítógépes Hálózatok 2008

6. Adatkapcsolati réteg – utólagos hibajavítás, csúszó ablakok, MAC, Statikus multiplexálás, (slotted) Aloha

Egyszerű simplex protokoll nyugtákkal

- Simplex üzemmód: csomagok küldése csak egyirányú
- A fogadó nyugtázza a küldő csomagjait (ehhez fél-duplex fizikai csatorna elegendő)
 - A küldő vár egy bizonyos ideig a nyugtára (acknowledgment -- ACK)
 - Ha az idő lejárt, újraküldi a csomagot
- Első megoldási kísérlet:

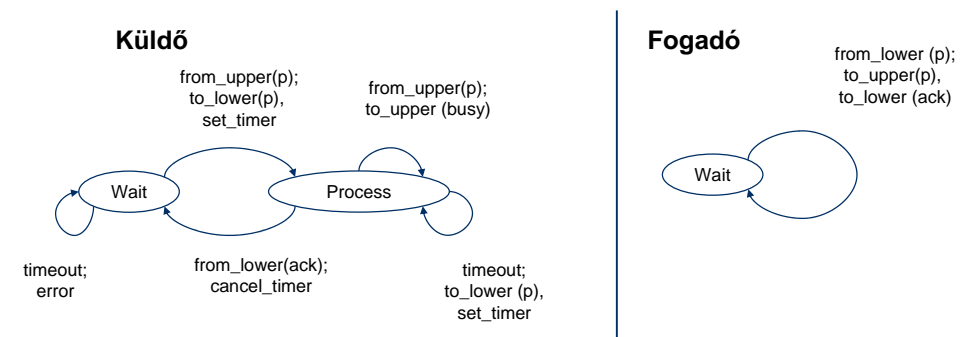


Elemzés

- Problémák
 - A felső réteg gyorsabban küldi a csomagokat, mint ahogy a nyugták megérkeznek
- Mi történik, ha nyugták elvesznek

2. Kísérlet

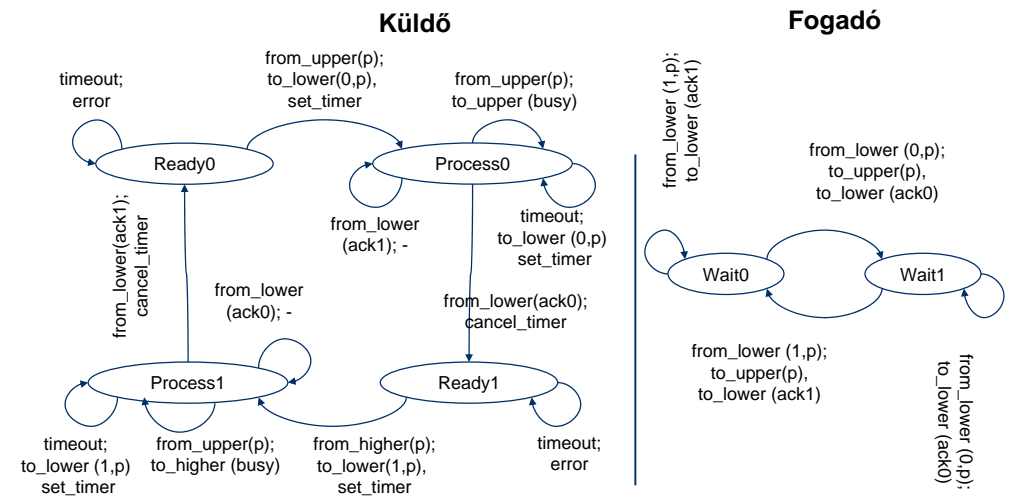
- Az első probléma megoldása
 - Egy csomag a másik után



A 2. probléma (duplikátumok)

- A küldő nem tud különbséget tenni elveszett csomag és elveszett nyugta között
 - Újra kell küldeni a csomagot
- A fogadó nem tud különbséget tenni egy csomag és egy régi csomag redundáns másolata között
 - További információ szükséges
- Ötlet:
 - Minden csomagot ellátunk egy **sorszámmal** (sequence number), hogy a fogadónál az azonosítás lehetséges legyen
 - Minden csomag fejléce tartalmaz sorszámot
 - Itt: csak 0 vagy 1
- Szükséges a csomagban és a nyugtában
 - A nyugta az utolsó hibátlanul fogadott csomag sorszámát tartalmazza (tisztán konvenció)

3. kísérlet: nyugta és sorszám

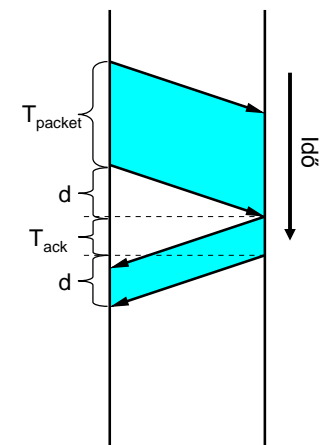


3. kísérlet: alternáló bit protokoll (Alternating Bit Protocol)

- A 3. kísérlet egy zajos csatorna fölötti megbízható protokoll korrekt implementációja
 - Alternating Bit Protokoll
 - Az „Automatic Repeat reQuest (ARQ)” protokollok közé tartozik
 - Folyamfelügyelet egy egyszerű formáját is tartalmazza
- Egy nyugta két feladata
 - nyugtázni, hogy egy csomag megérkezett
 - engedélyezni egy új csomag küldését

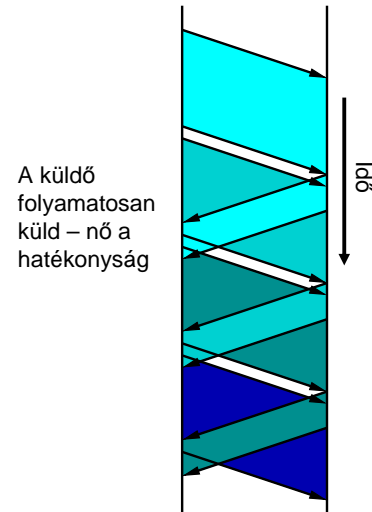
Alteráló bit protokoll -- hatékonyság

- Hatékonyság η a következő két érték arányaként definiált:
 - az idő, amely a küldéshez szükséges és
 - az idő, amely szükséges, amíg újra lehet küldeni
 - (hibamentes csatornán)
- $\eta = T_{\text{packet}} / (T_{\text{packet}} + d + T_{\text{ack}} + d)$
- Nagy delay esetén az alternáló bit protokoll nem hatékony



A hatékonyság javítása

- A csomagok folyamatos küldése növeli a hatékonyságot
 - több „outstanding” csomag (elküldött, de még nem nyugtázott) növeli a hatékonyságot
 - csomag „pipeline”
- Nem csak 1-bit-sorozatszámmal lehetséges

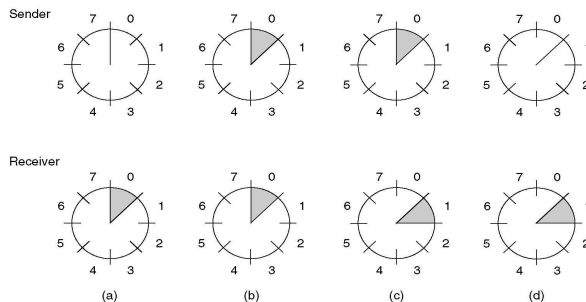


Csúszó ablak (sliding window)

- A sorozatszámok terét megnöveljük n bitre, azaz 2^n sorozatszámra
- Nem mind használható fel ugyanabban az időben
 - az Alternating Bit Protocol-ban sem lehetséges
- **„Csúszó ablakok” (sliding windows)** a küldőnél és a fogadónál kezelik ezt a problémát
 - Küldő: küldő-ablak
 - Sorozatszámok olyan sorozata, amelyek egy adott időben elküldhetők
 - Fogadó: fogadó-ablak
 - Sorozatszámok olyan sorozata, melyet a fogadó egy adott időpillanatban hajlandó elfogadni
- Az ablakok mérete lehet fix vagy időben dinamikusan változtatható
- Az ablakméret folyamfelügyeletet tesz lehetővé

Példa

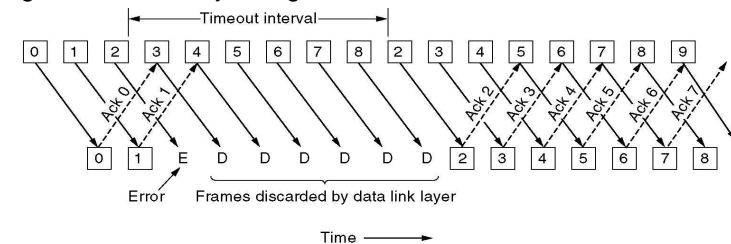
- „Sliding Window” példa $n=3$ és fix ablakméret = 1 esetén
- A küldő itt mutatja a még nem nyugtázott sorozatszámokat
 - Ha a még nem nyugtázott keretek (frame) száma ismert, akkor ez ekvivalens az előző főlíán definiált a küldő-ablakkal



- Kezdetben: mielőtt bármit küldenénk
- Az első frame küldése után 0 sorozatszámmal
- Az első frame fogadása után
- Az első nyugta fogadása után

Átviteli hiba és a fogadó-ablak

- Feltételeink:
 - Az adatátviteli rétegnek minden frame-et helyesen és **helyes sorrendben** kell átvinni
 - A küldő hatékonyság növeléséhez pipeline technikát használva küldi a csomagokat
- Csomagvesztés esetén: Ha a fogadó-ablakméret = 1, a következő csomagokat mind eldobja a fogadó



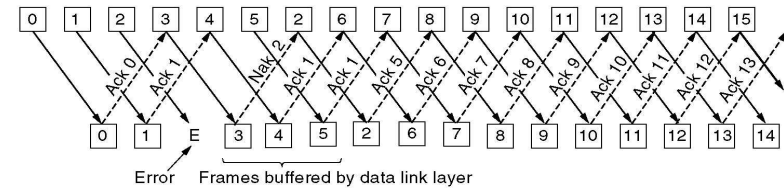
Go-back-N

- Ha a fogadó-ablakméret = 1, akkor a fogadó nem tudja feldolgozni azokat a frame-eket, melyek egy elveszett (vagy hibás) frame-et követnek
 - Nem tudja azokat nyugtázni, mert csak egy nyugtát küld az utolsó helyesen fogadott csomagról
- A küldőnél lejár a várakozási idő a nyugtára: "Timeout"
 - Minden frame-et, amit az utolsó nyugtázott frame után küldött, újra kell küldeni
 - "Go-back-N" Frames!
- Kritika
 - Az átviteli médium pazarlása
 - A fogadónál viszont nagyon egyszerű a feldolgozás

Szelektív ismétlés (Selective Repeat)

- Tegyük fel, hogy a fogadó tudja pufferelni a csomagokat, amelyek a közbelső időben érkeztek
 - azaz a fogadó-ablakméret > 1

● Példa



- A fogadó értesíti a küldőt a hiányzó csomagról negatív nyugtával
- A küldő elküldi a hiányzó frame-eket szelektíven (**selective repeat**)
- Amikor a hiányzó frame megérkezik, minden frame-et (a helyes sorrendben) átad a fogadó a hálózati rétegnek

Duplex-operáció és „hátizsák” technika (piggybacking)

- Simplex
 - Információ küldés egy irányba
- Duplex
 - Információ küldés mindkét irányba
- Eddig:
 - Simplex interfész a magasabb réteghez (hálózati réteghez)
 - (Fél-)Duplex interfész az alacsonyabb réteghez (fizikai réteghez)
- Mi kell akkor, ha az interfész a magasabb réteghez duplex
 - Nyugta és adatcsomagok elkülönítve mindkét irányban
 - Vagy: **hátizsák technika** (általánosan használt)
 - A nyugtát az ellentétes irányba küldött adat-frame fejlécébe tesszük (**piggybacking**)

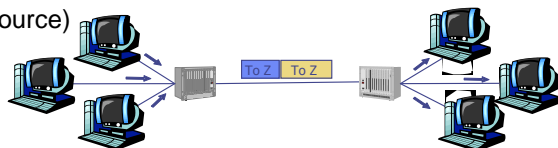
Mediumhozzáférés (Medium Access Control -- MAC) alréteg az adatkapcsolati rétegben

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

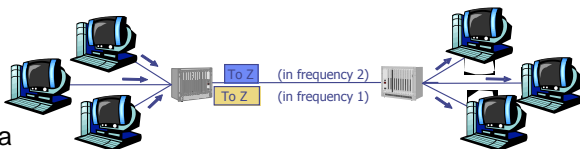
Statikus multiplexálás

- Adott egy link (erőforrás / ressource)

- A kommunikációs kapcsolatokhoz fix időegységeket (TDM) / frekvenciasávot (FDM) / csatornákat rendelünk



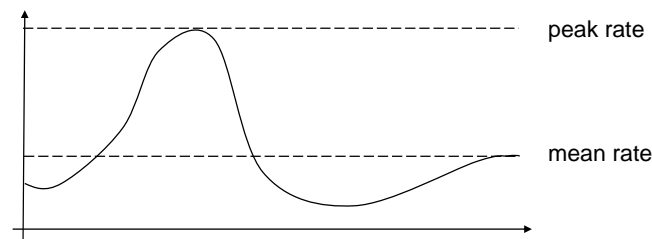
- Ez akkor jó megoldás, ha
 - fix adatráták vannak és a sávzélességet annak megfelelően osztjuk csatornákra
 - A források a vezetéket jól kihasználják



Löketszerűen érkező adatok (bursty traffic)

- Probléma: **bursty traffic**

- Definíció: nagy különbség a forgalom **csúcs rátája (peak rate)** és az **átlagos rátája (mean or average rate)** között
- Számítógép-hálózatokban peak rate / mean rate = 1000/1 nem szokatlan

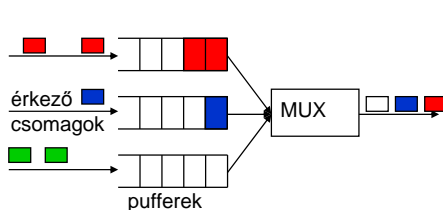
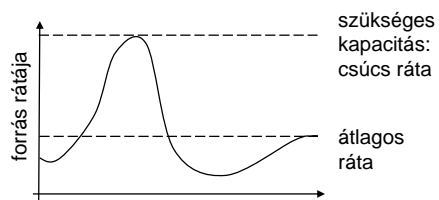


Löketszerűen érkező adatok és statikus multiplexálás

A linknek / csatornának statikus multiplexálás esetén vagy ... vagy ...

- vagy elegendően nagy kapacitásúnak kell lenni, hogy a csúcs rátát kezelni tudja
 - Pazarlás, mert az átlagos ráta nem használja ki a csatornát

- vagy az átlagos rátára alapozva kell dimenzionálni
 - akkor pufferek (queue) szükségesek
 - mi lesz a csomag késéssel (delay)?



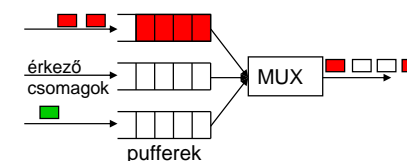
„Bursty traffic” és statikus multiplexálás – Késés (delay)

- Kiinduló helyzet:

- nincs multiplexálás (queue van)
- egy adatforrás p (bits/s) átlagos rátával
- a link kapacitása C bits/s
- a késés T

- Statikus multiplexálás esetén

- Osszuk az adatforrást N egyforma adatforrássá (mindegyik átlagos rátája p/N).
- Statikusan multiplexáljuk azokat ugyanazon a linken
- Ekkor a késés (lényegében): $T_{TDM, FDM} = N T$
 - (p/N átlagos küldési ráta és C/N kiszolgálási ráta \rightarrow sorok hossza?)
- Statikus multiplexálás megnöveli a csomagok késését az N -szeresére
- Ennek az oka: néhány csatorna sokszor „üres” (idle)



MAC alréteg

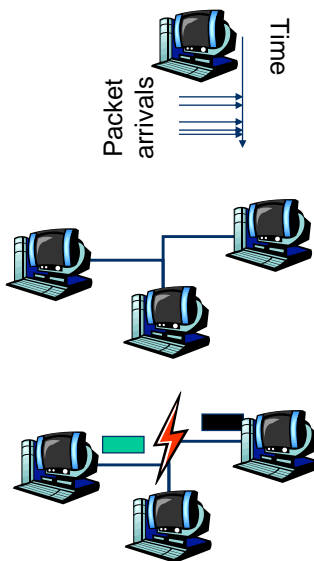
- Statikus Multiplexálás
- **Dinamikus csatorna foglalás**
 - Kollízió alapú protokollok
 - Verseny-mentes protokollok (contention-free)
 - Protokollok korlátozott versennyel (limited contention)
- Az Ethernet példája

Dinamikus csatorna foglalás – MAC

- Statikus multiplexálás nem megfelelő löketszerű adatforgalom kezelésére
 - Telefon hálózatok forgalma nem löketszerű, számítógépes hálózatoké az
- Alternatíva: A csatorna/link/erőforrás hozzárendelése ahhoz a forrshoz aki éppen adatot akar küldeni
 - Dinamikus csatorna foglalás (channel allocation)
 - az erőforrás fix részének hozzárendelése helyett
- Szabályozni kell a médium hozzáférést:
 - Médium hozzáférés protokoll (Medium Access Control protocol - MAC) szükséges

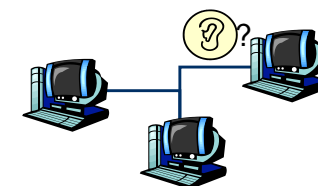
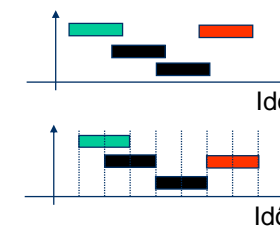
A dinamikus csatornafoglalás modellje

- **N állomás** (vagy N terminal)
 - N független állomás használja az adott erőforrást
 - Egy lehetséges **terhelés modell**: annak a valószínűsége, hogy egy állomás Δt intervallumban csomagot generál: $\lambda \Delta t$, ahol $\lambda = \text{konstans}$
- **Egy csatorna**
 - Az összes állomás részére együttesen egy csatorna áll rendelkezésre
 - A csatornán kívül semmilyen más lehetőség nincs kommunikálni egymással
- **Kollízió modell** (ütközés)
 - Egy időben csak egy frame vihető át eredményesen
 - Ha két (vagy több) frame időben átfedi egymást, akkor azok ütköznek és mindkettő szétrombolódik
 - Egy állomás se tudja fogadni egyik frame-et sem
 - Megjegyzés: ez alól a szabály alól van néha kivétel (pl. CDMA)



Modellek

- **Időmodellek**
 - Folytonos
 - Átvitel minden időben kezdődhet (nincs központi óra)
 - Diszkrét (Slotted time)
 - Az idő-tengely darabokra (slots) van osztva
 - Átvitel csak egy slot határán kezdődhet
 - Egy slot lehet üres (idle), vagy sikeresen átvitt, vagy kollíziót tartalmazó
- **Vivő-érzékelés (Carrier Sensing)**
 - Az állomások képesek felismerni, hogy éppen egy más állomás használja-e a csatornát
 - Nem feltétlenül megbízhatóan (pl. egy éppen kezdődő átvitelnél)
 - Ha a csatorna foglalt (busy), nem indít az állomás átvitelt

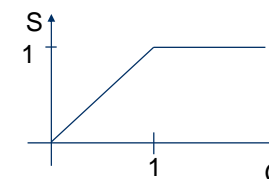


A hatékonyság mérése

- A csatornafoglalás hatékonyságának mértékei
- **Átvitel** (throughput)
 - Csomagok száma időegységenként
 - Különösen nagy terhelés esetén fontos
- **Késés** (delay)
 - Egy csomag átviteléhez szükséges idő
 - Alacsony terhelés esetén
- **Fairness**
 - Minden állomást egyenlőként kezelünk
 - Az átvitel és a késés körülbelül egyforma legyen az állomásokon

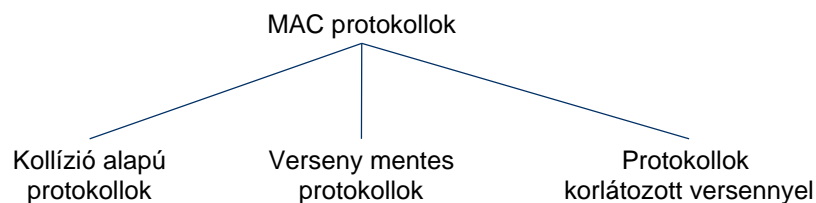
Átvitel és a feldolgozandó terhelés (offered load)

- **Feldolgozandó terhelés (offered load) G**
 - A csomagok száma csomag-időegységenként, amit a protokollnak kezelnie kell
 - $G > 1$: túlterhelés
- **Ideális protokoll**
 - Amíg $G < 1$, akkor az átvitel S egyenlő G -vel
 - Ha $G \geq 1$, akkor $S = 1$
- És: konstans kis késés tetszőlegesen sok állomás esetén is



Lehetséges MAC-protokollok

- Fő megkülönböztetés: Megenged-e a protokoll kollíziót?
 - Rendszer döntés
 - A feltétlen kollízió-elkerülés a hatékonyság csökkenésével járhat



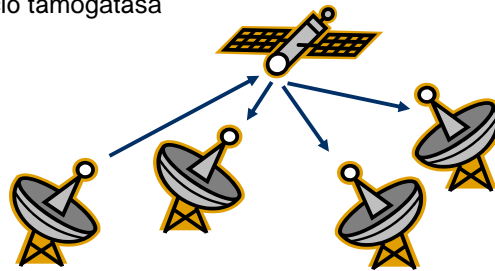
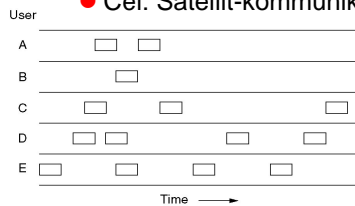
Rendszer, amelyben kollízió történhet:
Contention System (verseny rendszer)

MAC alréteg

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

ALOHA

- Algoritmus:
 - Amikor egy csomag kész, azonnal átvitelre kerül
- Történet:
 - 1985 by Abrahmson et al., University of Hawaii
 - Cél: Satellit-kommunikáció támogatása



A csomagok tetszőleges időben kerülnek átvitelre

ALOHA – Elemzés

- Előny
 - Egyszerű
 - Koordináció nem szükséges
- Hátrányok
 - Kollíziók
 - A küldő nem teszteli a csatorna állapotát
 - A küldőnek nincs direkt módszere arra, hogy megtudja, hogy eredményes volt-e az átvitel
 - Nyugták (ACK) szükségesek
 - A nyugták szintén ütközhetnek

ALOHA – Hatékonyság

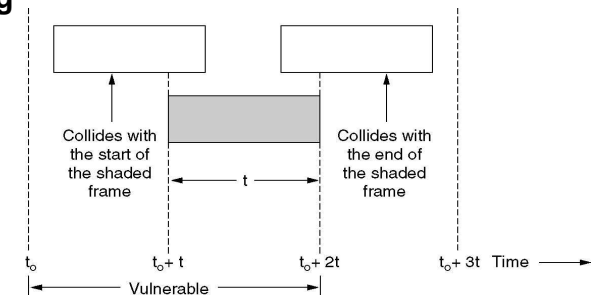
- Tegyük fel, hogy a csomagok létrehozása **Poisson-folyamat**:
 - „Végtelen” sok állomás, melyek egyformán, függetlenül viselkednek
 - Minden csomag egyforma hosszú, annak átviteléhez egységnyi idő kell
 - Az idő két küldési kísérlet között exponenciális eloszlású
 - Legyen G a küldési kísérletek számának várható értéke egységnyi idő alatt (egységnyi idő = egy csomag átviteléhez szükséges idő)
 - Ekkor:

$$P[k \text{ küldési kísérlet } t \text{ idő alatt}] = \frac{(Gt)^k}{k!} e^{-Gt}$$

- Ahhoz hogy sikeres átvitelt hajtsunk végre, nem szabad hogy kollízió lépjen fel egy másik csomaggal
- Mi ennek a valószínűsége?

ALOHA – Hatékonyság

- Egy X csomag ütközik, ha
 - egy csomag nem ért véget, amikor X indul
 - egy csomag kicsivel X vége előtt indul



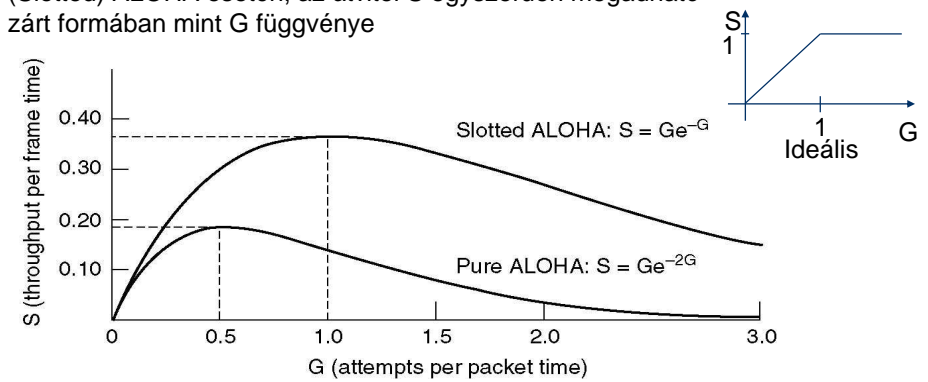
- Azaz, egy csomagátvitel akkor sikeres, ha két egységnyi időben nincs másik csomagátviteli kísérlet
 - Valószínűség: $P_0 = P(0 \text{ csomag } 2 \text{ egységnyi időben}) = e^{-2G}$
 - Maximális átvitel $S(G) = G * P_0 = G * e^{-2G}$
 - Optimum $G = 0,5$ -nál: $S = 1/(2e) \approx 0,184$

Egy javítás: Slotted ALOHA

- ALOHA problémája: a csomag „sebezhetőségi” ideje hosszú (2 időegység)
- Csökkentjük idődarabok (slot) bevezetésével – átvitel csak egy slot elején kezdődhet
 - Feltesszük, hogy a slot-ok szinkronizálása „valahogy” rendelkezésre áll
- Eredmény: Sebezhetőségi idő feleződik, az átvitel megduplázódik
 - $S(G) = Ge^{-G}$
 - Optimum $G=1$ -nél: $S=1/e$

Hatékonyág a feldolgozandó terhelés függvényében

- (Slotted) ALOHA esetén, az átvitel S egyszerűen megadható zárt formában mint G függvénye



- Az átvitel összezuhan, ha nő a terhelés!