

Számítógépes Hálózatok

5. Előadás: Adatkapcsolati réteg 2.rész

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

Elméleti alapok

- Tegyük fel, hogy a keret m bitet tartalmaz. (üzenet bitek)
- A redundáns bitek száma legyen r . (ellenőrző bitek)
- A küldendő keret tehát $n=m+r$ bit hosszú. (kódszó)

Hamming távolság

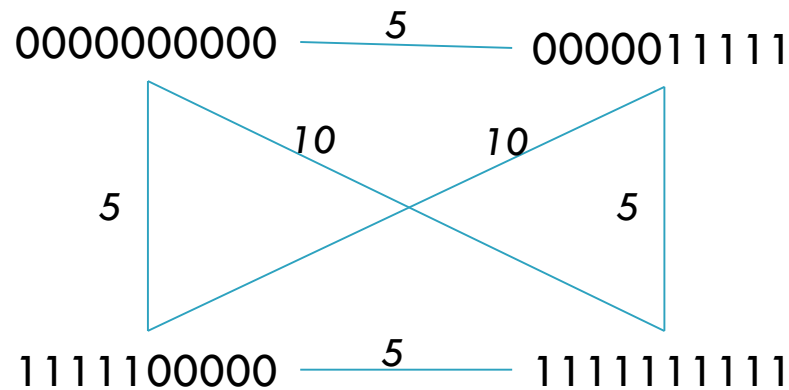
- Az olyan bitpozíciók számát, amelyeken a két kódszóban különböző bitek állnak, a két kódszó Hamming távolságának nevezzük.
 - ▣ Jelölés: $d(x,y)$
- Legyen S egyenlő hosszú bitszavak halmaza, ekkor S Hamming távolsága az alábbi:

$$d(S) := \min_{x,y \in S \wedge x \neq y} d(x,y)$$

- ▣ Jelölés: $d(S)$
- A Hamming távolság egy metrika.

Példa Hamming távolságra

- Legyen $S = \{0000000000, 0000011111, 1111100000, 1111111111\}$.
- Mi lesz a halmaz Hamming távolsága?
 - ▣ $d(S) = 5$



Hamming távolság használata

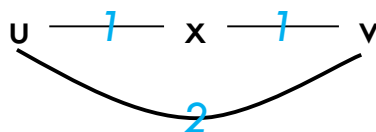
S halmaz legyen a megengedett azonos hosszú kódszavak halmaza.

$d(S)=1$ esetén

- nincs hibafelismerés
- megengedett kódszóból megengedett kódszó állhat elő 1 bit megváltoztatásával

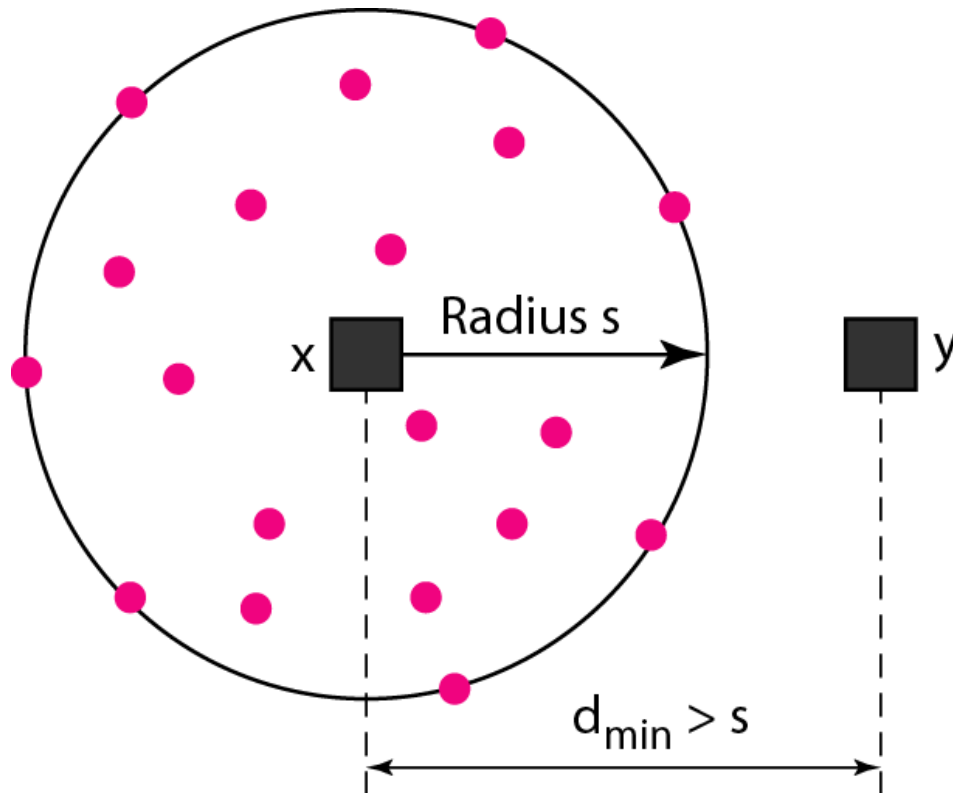
$d(S)=2$ esetén

- ha az u kódszóhoz létezik olyan x megengedett kódszó, amelyre $d(u, x) = 1$, akkor hiba történt.
- Feltéve, hogy az u és v megengedett kódszavak távolsága minimális, akkor a következő összefüggésnek teljesülnie kell: $2 = d(u, v) \leq d(u, x) + d(x, v)$.
- Azaz egy bithiba felismerhető, de nem javítható.



Hiba felismerés

d bithiba felismeréséhez legalább $d+1$ Hamming távolságú kód szükséges.



Legend



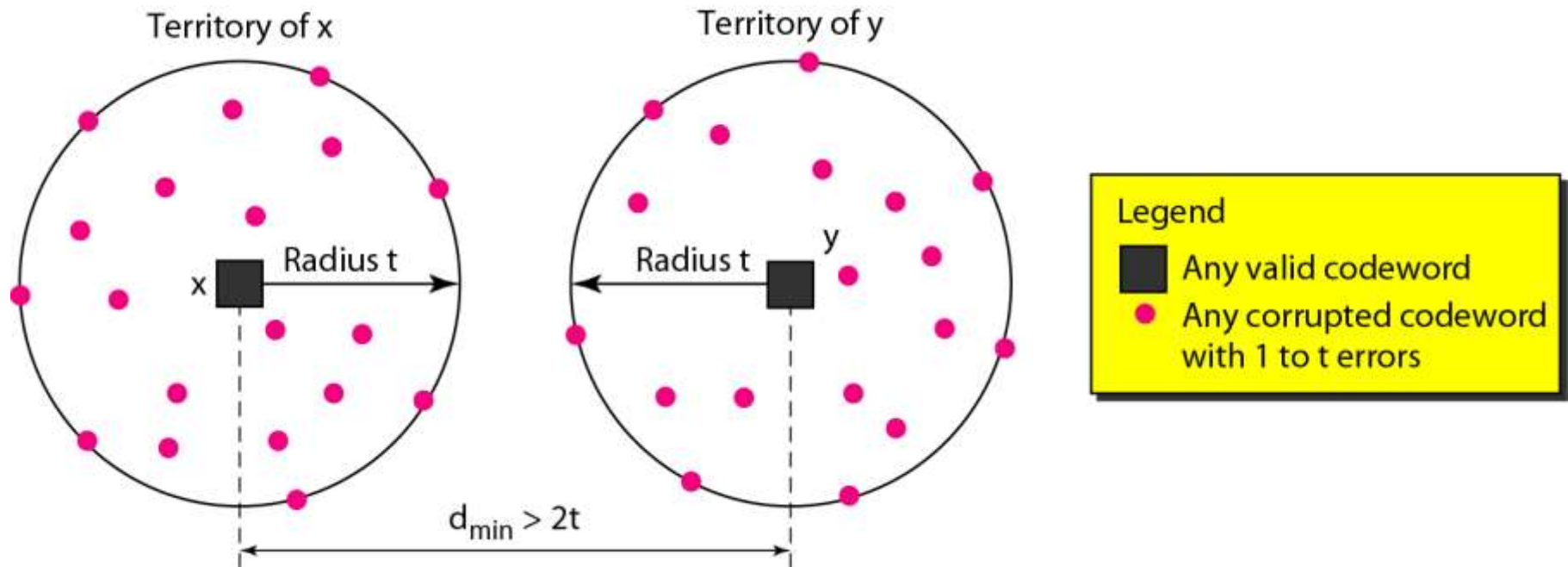
Any valid codeword



Any corrupted codeword
with 0 to s errors

Hiba javítás

d bithiba javításához legalább $2d+1$ Hamming-távolságú kód szükséges.



Hamming korlát bináris kódkönyvre 1/3

TÉTEL

Minden $C \subseteq \{0,1\}^n$ kód, ahol $d(C) = k$ ($\in \mathbb{N}_+$). Akkor teljesül az alábbi összefüggés:

$$|C| \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{n}{i} \leq 2^n$$

BIZONYÍTÁS

1. Hány olyan bitszó létezik, amely egy tetszőleges $x \in C$ kódszótól pontosan $i \in \mathbb{N}_+$ távolságra helyezkedik el?
 - ▣ Pontosan $\binom{n}{i}$ lehetőség van.
2. Hány olyan bitszó létezik, amely egy tetszőleges $x \in C$ kódszótól legfeljebb $\lfloor \frac{k-1}{2} \rfloor$ távolságra helyezkedik el?
 - Pontosan $\sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{n}{i}$ lehetőség van.

Hamming korlát bináris kódkönyvre 2/3

3. Lássuk be, hogy egy tetszőleges $x \in \{0,1\}^n$ bitszóhoz legfeljebb egy legális $u \in C$ kódszó létezhet, amelyre $d(x, u) \leq \frac{k-1}{2}$ teljesül.
- ▣ Indirekt tegyük fel, hogy létezhet két legális kódszó is a C kódkönyvben, jelölje őket u_1 és u_2 . Ekkor viszont az alábbi két feltétel együttesen teljesül:

$$d(x, u_1) \leq \frac{k-1}{2} \text{ és } d(x, u_2) \leq \frac{k-1}{2}$$

- ▣ Mi a két kódszó távolsága?

$$d(u_2, u_1) \leq d(u_2, x) + d(x, u_1) \leq \frac{k-1}{2} + \frac{k-1}{2} = k-1$$

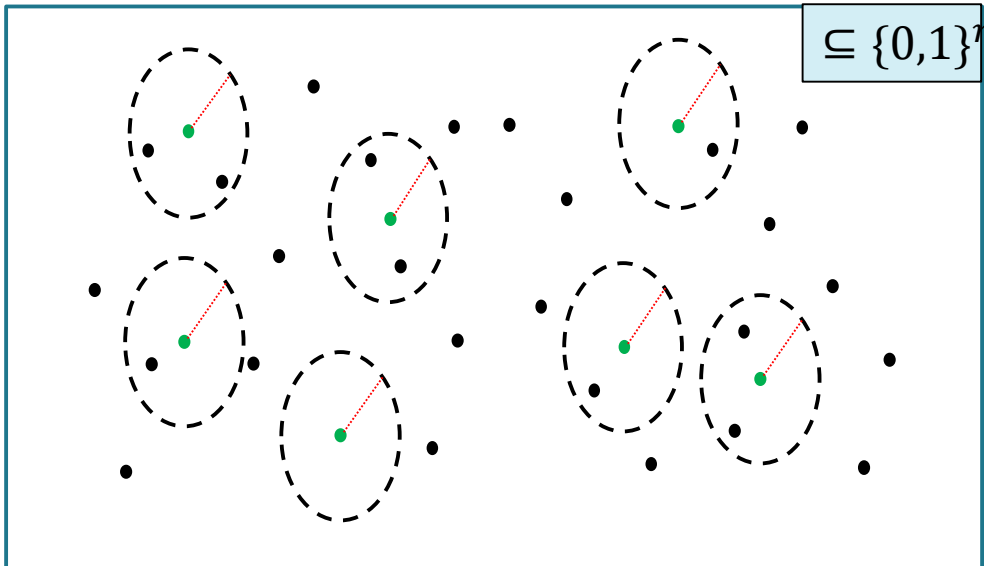
- ▣ Ez viszont ellentmond annak hogy a kódkönyv Hamming távolsága k , azaz az indirekt feltevésünk volt hibás. Vagyis tetszőleges bitszóhoz legfeljebb egy legális kódszó létezhet, amely a kódkönyv minimális távolságának felénél közelebb van a bitszóhoz.

Hamming korlát bináris kódkönyvre 3/3

4. A kódszavak $\frac{k-1}{2}$ sugarú környezetében található bitszavak egymással diszjunkt halmazainak uniója legfeljebb az n -hosszú bitszavak halmazát adhatja ki. Vagyis formálisan:

$$|C| \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{n}{i} \leq 2^n$$

□



JELMAGYARÁZAT

- Kódszó
- Bitszó, amely nem kódszó

Hibafelismerés és javítás Hamming távolsággal

Hibafelismerés

- d bit hiba felismeréséhez a megengedett keretek halmazában legalább $d+1$ Hamming távolság szükséges.

Hibajavítás

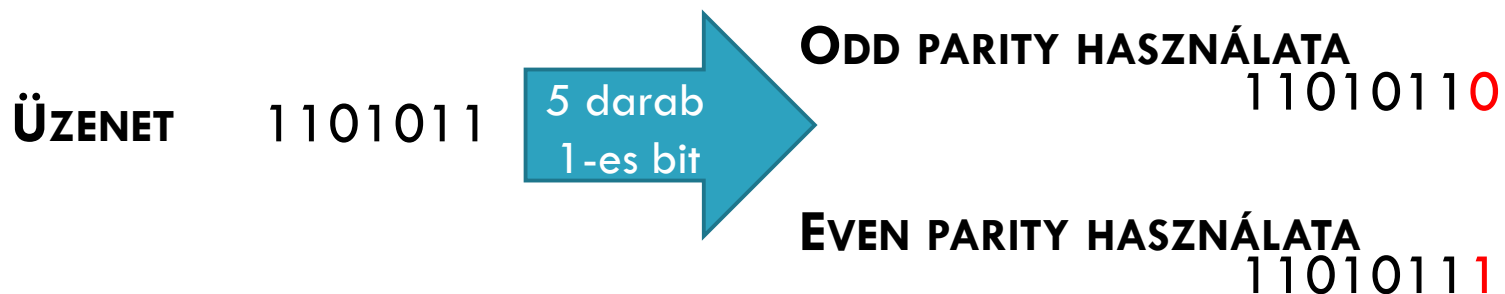
- d bit hiba javításához a megengedett keretek halmazában legalább $2d+1$ Hamming távolság szükséges

Definíciók

- Egy $S \subseteq \{0,1\}^n$ kód rátája $R_S = \frac{\log_2 |S|}{n}$. (a hatékonyságot karakterizálja)
- Egy $S \subseteq \{0,1\}^n$ kód távolsága $\delta_S = \frac{d(S)}{n}$. (a hibakezelési lehetőségeket karakterizálja)
- A jó kódoknak a rátája és a távolsága is nagy.

Újra a paritás bit használata 1/4

- a paritásbitet úgy választjuk meg, hogy a kódszóban levő 1-ek száma páros (vagy páratlan)
 - ▣ **Odd parity** – ha az egyesek száma páratlan, akkor 0 befűzése; egyébként 1-es befűzése
 - ▣ **Even parity** – ha az egyesek száma páros, akkor 0 befűzése; egyébként 1-es befűzése



Paritás bit használata 2/4

Egy paritást használó módszer (*Hamming*)

- a kódszó bitjeit számozzuk meg 1-gyel kezdődően;
- 2 egészhatvány sorszámú pozíciói lesznek az ellenőrző bitek, azaz 1,2,4,8,16,...;
- a maradék helyeket az üzenet bitjeivel töltjük fel;
- mindegyik ellenőrző bit a bitek valamilyen csoportjának a paritását állítja be párosra (vagy páratlanra)
- egy bit számos paritásszámítási csoportba tartozhat:
 - k pozíciót írjuk fel kettő hatványok összegeként, a felbontásban szereplő ellenőrző pozíciók ellenőrzik a k -adik pozíciót
 - Példa: $k=13$ -ra $k=1+4+8$, azaz az első, a negyedik illetve a nyolcadik ellenőrző bit fogja ellenőrizni

Paritás bit használata - példa 3/4

- Az *ASCII* kód 7 biten ábrázolja a karaktereket
- A példában *EVEN PARITY*-t használunk

ÜZENET BITEK KÓDSZÓBAN LÉVŐ POZÍCIÓNAK FELBONTÁSAI

- $3 = 1 + 2$
- $5 = 1 + 4$
- $6 = 2 + 4$
- $7 = 1 + 2 + 4$
- $9 = 1 + 8$
- $10 = 2 + 8$
- $11 = 1 + 2 + 8$

ASCII karakter	ASCII decimális	Üzenet forrás bitjei	Az előállt kódszavak
E	69	1000101	10100000101
L	76	1001100	10110011100
T	84	1010100	00110101100
E	69	1000101	10100000101
	32	0100000	10001100000
I	73	1001001	11110011001
K	75	1001011	00110010011

Paritás bit használata 4/4

- a vevő az üzenet megérkezésekor 0-ára állítja a számlálóját, ezt követően megvizsgálja a paritás biteket, ha a k -adik paritás nem jó, akkor a számlálóhoz ad k -t
- Ha a számláló 0 lesz, akkor érvényes kódszónak tekinti a vevő a kapott üzenetet; ha a számláló nem nulla, akkor a hibás bit sorszámát tartalmazza, azaz ha például az első, a második és nyolcadik bit helytelen, akkor a megváltozott bit a tizenegyedik.

FOGADOTT *E* KARAKTER 10100100101

Számláló != 0

SZÁMLÁLÓ = 2 + 4

FOGADOTT *L* KARAKTER 11110011100

Számláló != 0

SZÁMLÁLÓ = 2

Hibajelző kódok

Hibajelző kódok

Polinom-kód, avagy ciklikus redundancia (CRC kód)

- Tekintsük a bitsorozatokat \mathbb{Z}_2 feletti polinomok reprezentációinak.

Polinom ábrázolása \mathbb{Z}_2 felett

$$p(x) = \sum_{i=0}^n a_i x^i = a_n x^n + \dots + a_1 x^1 + a_0 x^0, \text{ ahol } a_i \in \{0,1\}$$

- A számítás *mod 2* történik. (összeadás, kivonás, szorzás, osztás)
- reprezentálható az együtthatók $n+1$ -es vektorával, azaz (a_n, \dots, a_1, a_0)
- Például az ASCII „b” karakter kódja 01100010, aminek megfelelő polinom hatod fokú polinom

$$p(x) = 1 * x^6 + 1 * x^5 + 0 * x^4 + 0 * x^3 + 0 * x^2 + 1 * x^1 + 0 * x^0$$

- Az összeadás és a kivonás gyakorlati szempontból a logikai KIZÁRÓ VAGY művelettel azonosak.

$$\begin{array}{r} 11110000 \\ - 10100110 \\ \hline 01010110 \end{array}$$

$$\begin{array}{r} 10011011 \\ + 11001010 \\ \hline 01010001 \end{array}$$

CRC

Definiáljuk a $G(x)$ generátor polinomot (G foka r), amelyet a küldő és a vevő egyaránt ismer.

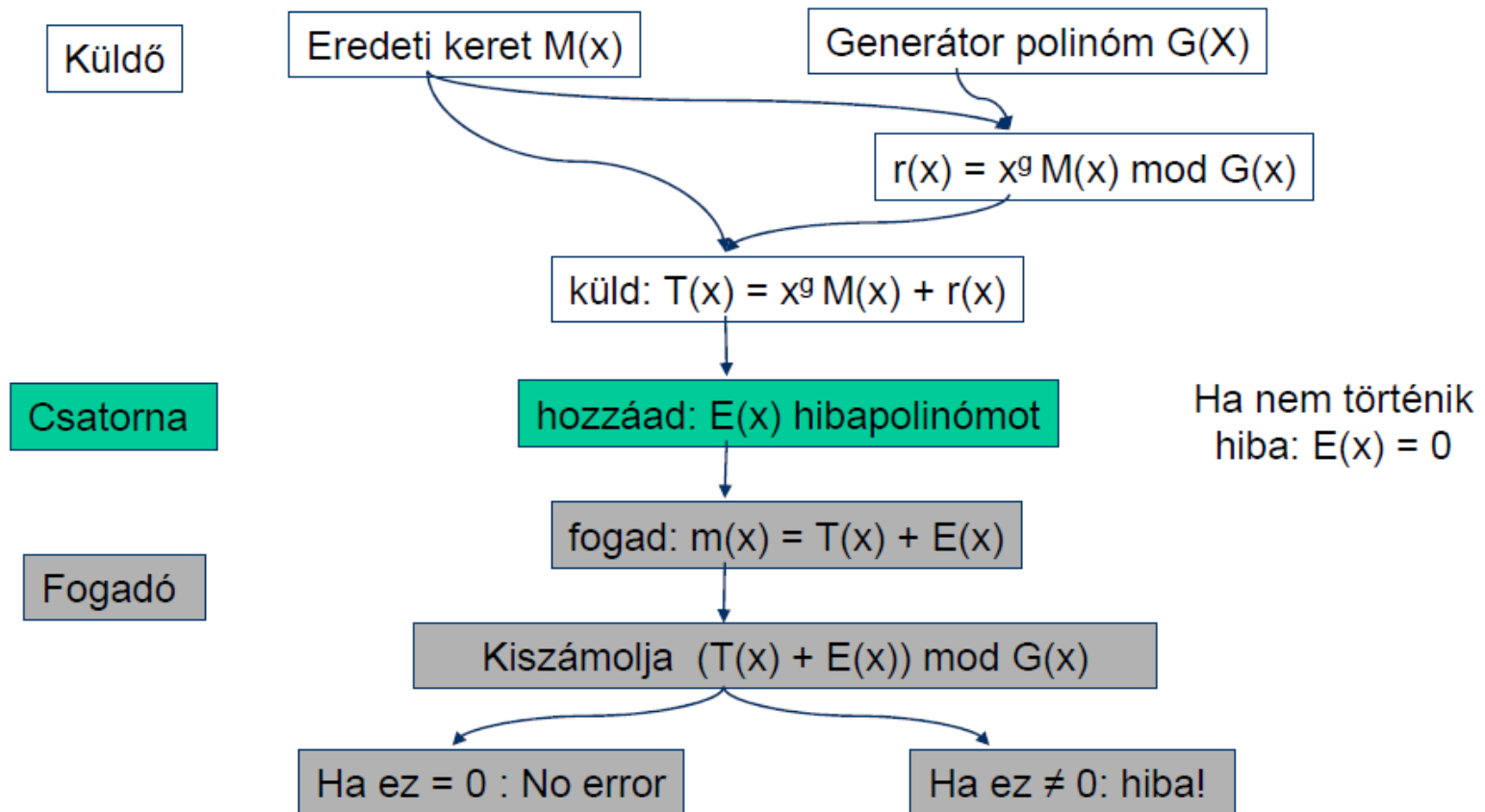
Algoritmus

1. Legyen $G(x)$ foka r . Fűzzünk r darab 0 bitet a keret alacsony helyi értékű végéhez, így az $m+r$ bitet fog tartalmazni és az $x^r M(x)$ polinomot fogja reprezentálni.
2. Osszuk el az $x^r M(x)$ tartozó bitsorozatot a $G(x)$ -hez tartozó bitsorozattal modulo 2.
3. Vonjuk ki a maradékot (mely mindig r vagy kevesebb bitet tartalmaz) az $x^r M(x)$ -hez tartozó bitsorozatból moduló 2-es kivonással. Az eredmény az ellenőrző összeggel ellátott, továbbítandó keret. Jelölje a továbbítandó keretnek megfelelő a polinomot $T(x)$.
4. A vevő a $T(x) + E(x)$ polinomnak megfelelő sorozatot kapja, ahol $E(x)$ a hiba polinom. Ezt elosztja $G(x)$ generátor polinommal.
 - Ha az osztási maradék, amit $R(x)$ jelöl, nem nulla, akkor hiba történt.

CRC áttekintés

18

□ Forrás: Dr. Lukovszki Tamás fóliái

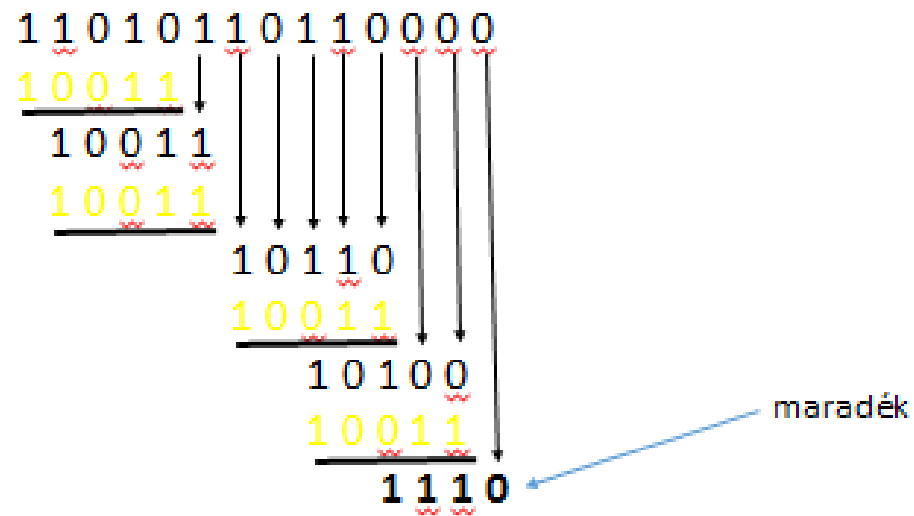


Példa CRC számításra

Keret: 1101011011

Generátor: 10011

A továbbítandó üzenet: 11010110111110



CRC áttekintés

- A $G(x)$ többszöröseinek megfelelő bithibákat nem ismerjük fel, azaz, ha $\exists j \in \mathbb{N}: E(x) = x^j G(x)$.
- $G(x)$ legmagasabb illetve legalacsonyabb fokú tagjának együtthatója mindig 1.

Hiba események

- $E(x) = x^i$, azaz i a hibás bit sorszáma, mivel $G(x)$ kettő vagy több tagból áll, ezért minden egybites hibát jelezni tud.
- $E(x) = x^i + x^j = x^j (x^{i-j} + 1)$ ($i > j$), azaz két izolált egybites hiba esetén.
 - ▣ $G(x)$ ne legyen osztható x -szel;
 - ▣ $G(x)$ ne legyen osztható $(x^k + 1)$ –gyel semmilyen maximális kerethossznál kisebb k -ra. (Pl. $x^{15} + x^{14} + 1$)
- Ha $E(x)$ páratlan számú tagot tartalmaz, akkor nem lehet $x+1$ többszöröse. Azaz, ha $G(x)$ az $x+1$ többszöröse, akkor minden páratlan számú hiba felismerhető
- Egy r ellenőrző bittel ellátott polinom-kód minden legfeljebb r hosszúságú csoportos hibát jelezni tud

CRC a gyakorlatban

- IEEE 802 által használt polinom az

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

- Néhány jó tulajdonságai a fenti polinomnak:

1. minden legfeljebb 32 bites hibacsomót képes jelezni,
2. minden páratlan számú bitet érintő hibacsomót tud jelezni.

Peterson és Brown (1961)

- Szerkeszthető egy egyszerű, léptető regiszteres áramkör az ellenőrző összeg hardverben történő kiszámítására és ellenőrzésére.

Adatkapcsolati réteg

22



- 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 címezés (pl. MAC address, IB address)
- Példák: Ethernet, Wifi, InfiniBand

Adatkapcsolati réteg

23



□ 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 akár 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

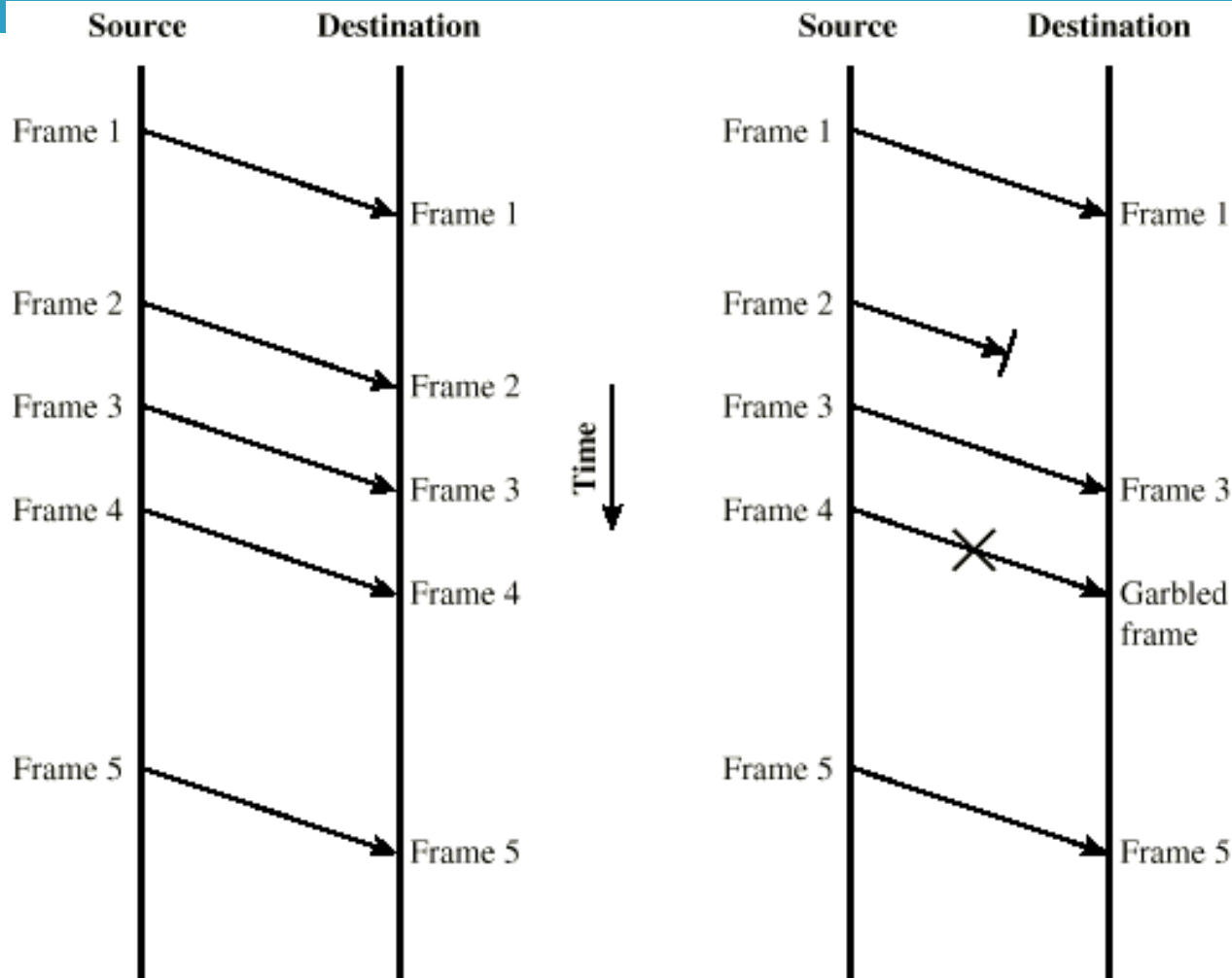
- mind az adó, mind a vevő hálózati rétegei mindig készen állnak;
- a feldolgozási időktől eltekintünk;
- végtelen puffer-területet feltételezünk;
- Az adatkapcsolati rétegek közötti kommunikációs csatorna sosem rontja vagy veszíti el a kereteket;

A protokoll

- résztvevők: *küldő* és *vevő*;
- nincs sem sorszámozás, sem nyugta;
- küldő végtelen ciklusban küldi kifelé 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

28



(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 puffereelés 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 pufferek és sorban állás sem);
- Az adatkapcsolati rétegek közötti kommunikációs csatorna hibázhat (keret megsérülése vagy elvesztése);

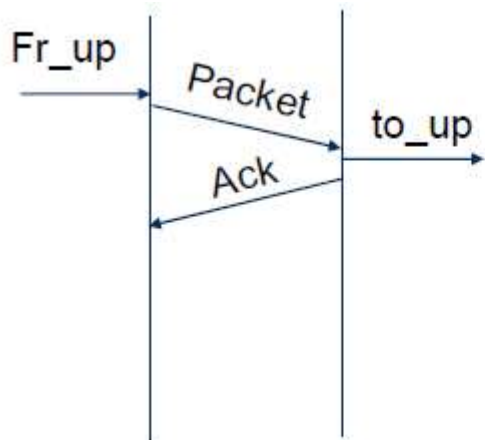
A protokoll

- 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 ismételt 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.

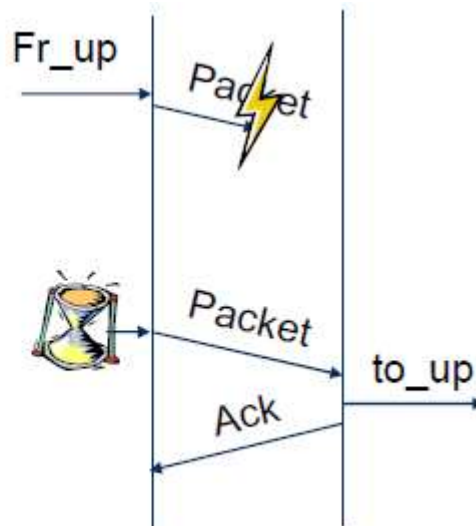
Következmény: duplikátumok lehetnek.

Mi is a probléma?

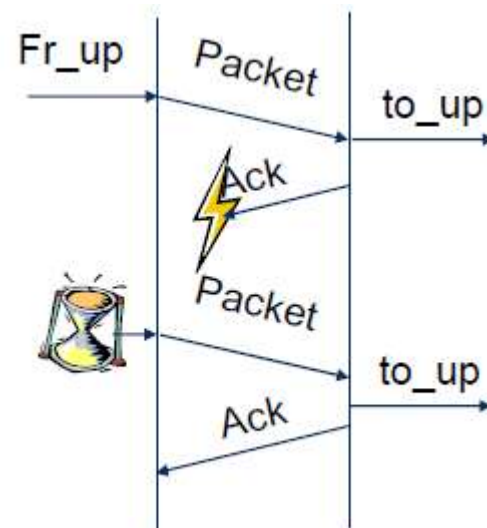
Általában



Csomagvesztés esetén

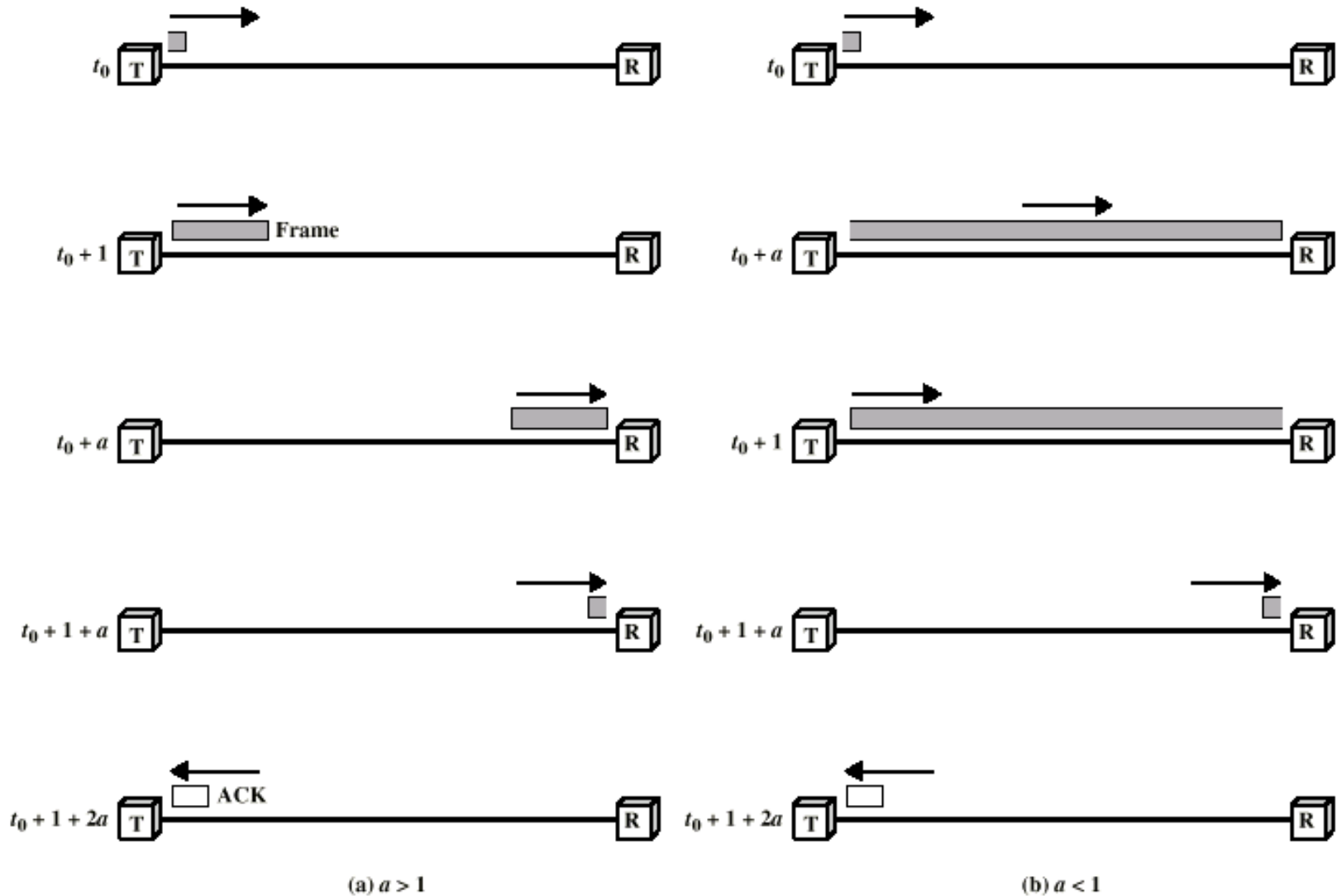


ACK vesztés esetén



Csatorna kihasználtság

32



Alternáló-bit protokoll (ABP)

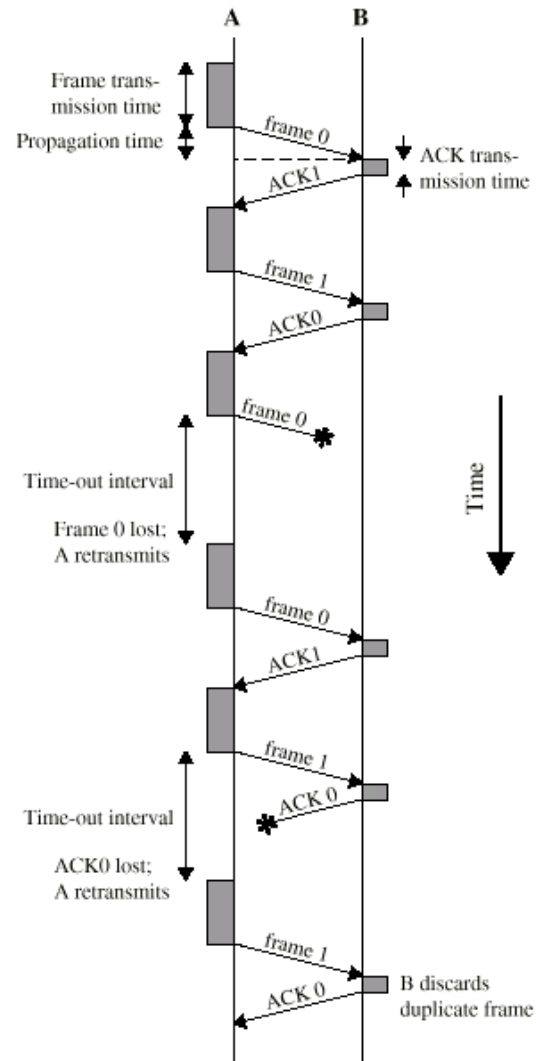
- ❑ **Megoldás:** sorszámok használata
- ❑ Mennyi sorszámra lesz szükség? $\{0,1\}$ elegendő

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.

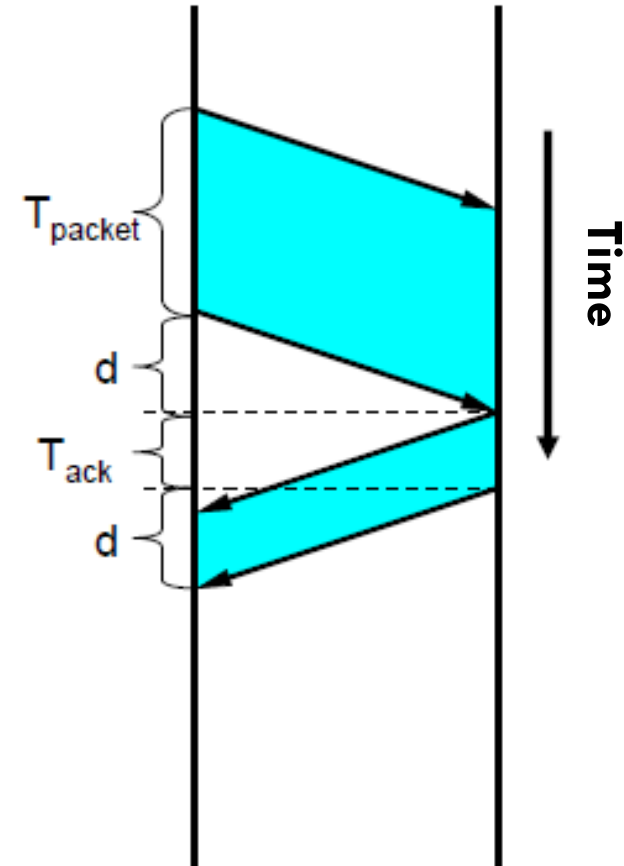
ABP

34



ABP – Csatorna kihasználtság

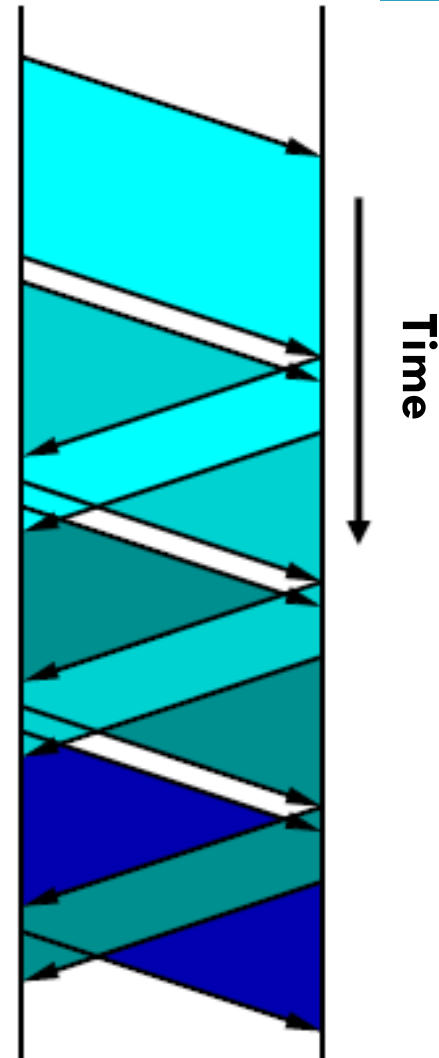
- Kihasználtság (η) a következő két elem aránya
 - ▣ A csomag elküldéséhez szükséges idő (T_{packet})
 - ▣ Az idő, ami a következő keret küldéséig eltelik
 - Az ábrán: ($T_{\text{packet}} + d + T_{\text{ack}} + d$)
- ABP esetén:
 - ▣ $\eta = T_{\text{packet}} / (T_{\text{packet}} + d + T_{\text{ack}}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

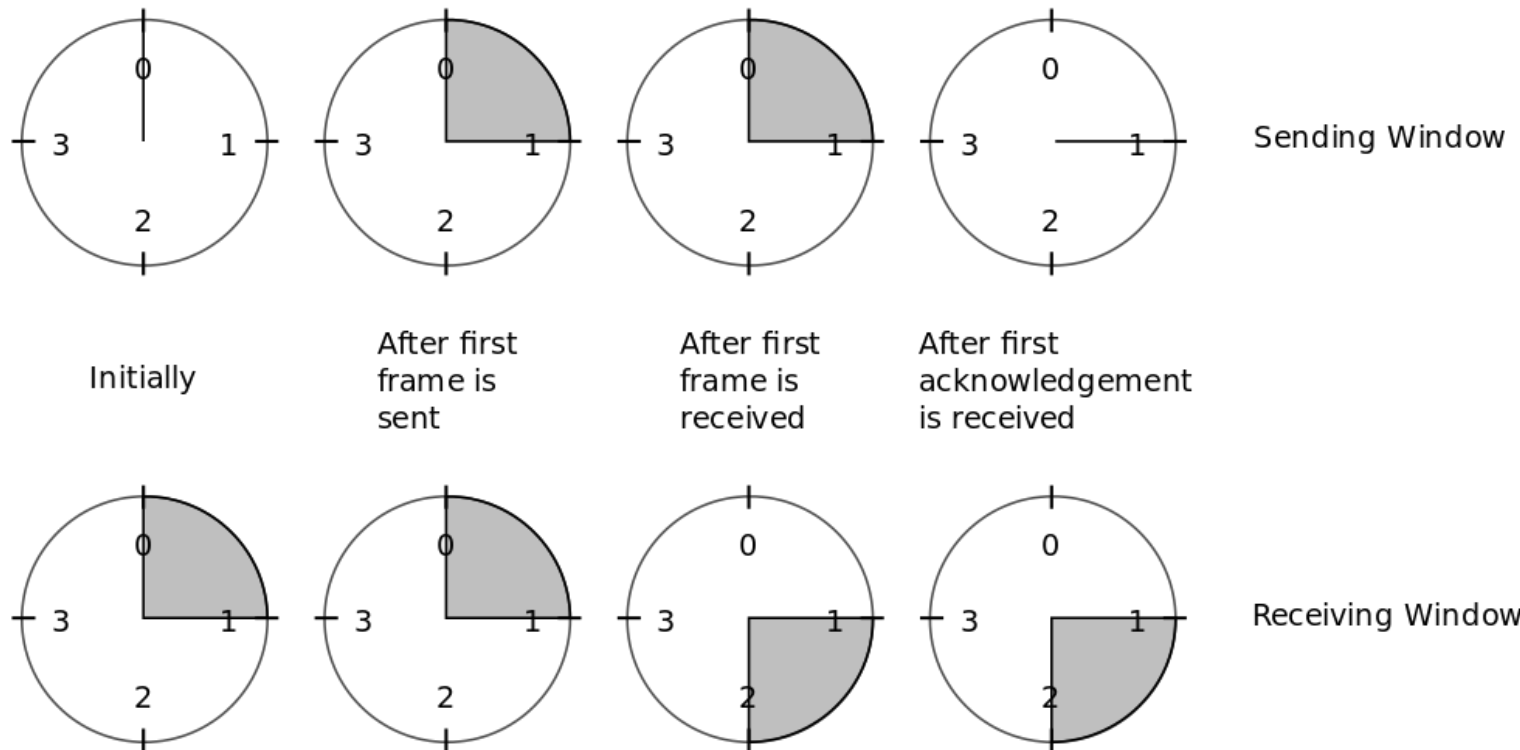
- Egy adott időpontban egyszerre több keret is átviteli állapotban lehet.
- A fogadó n keretnek megfelelő méretű puffert allokál.
- A küldőnek legfeljebb n , azaz ablak méretnyi, nyugtázatlan keretet küldése engedélyezett.
- A keret sorozatbeli pozíciója adja a keret címkéjét. (sorozatszám)

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.

Csúszó ablak

38



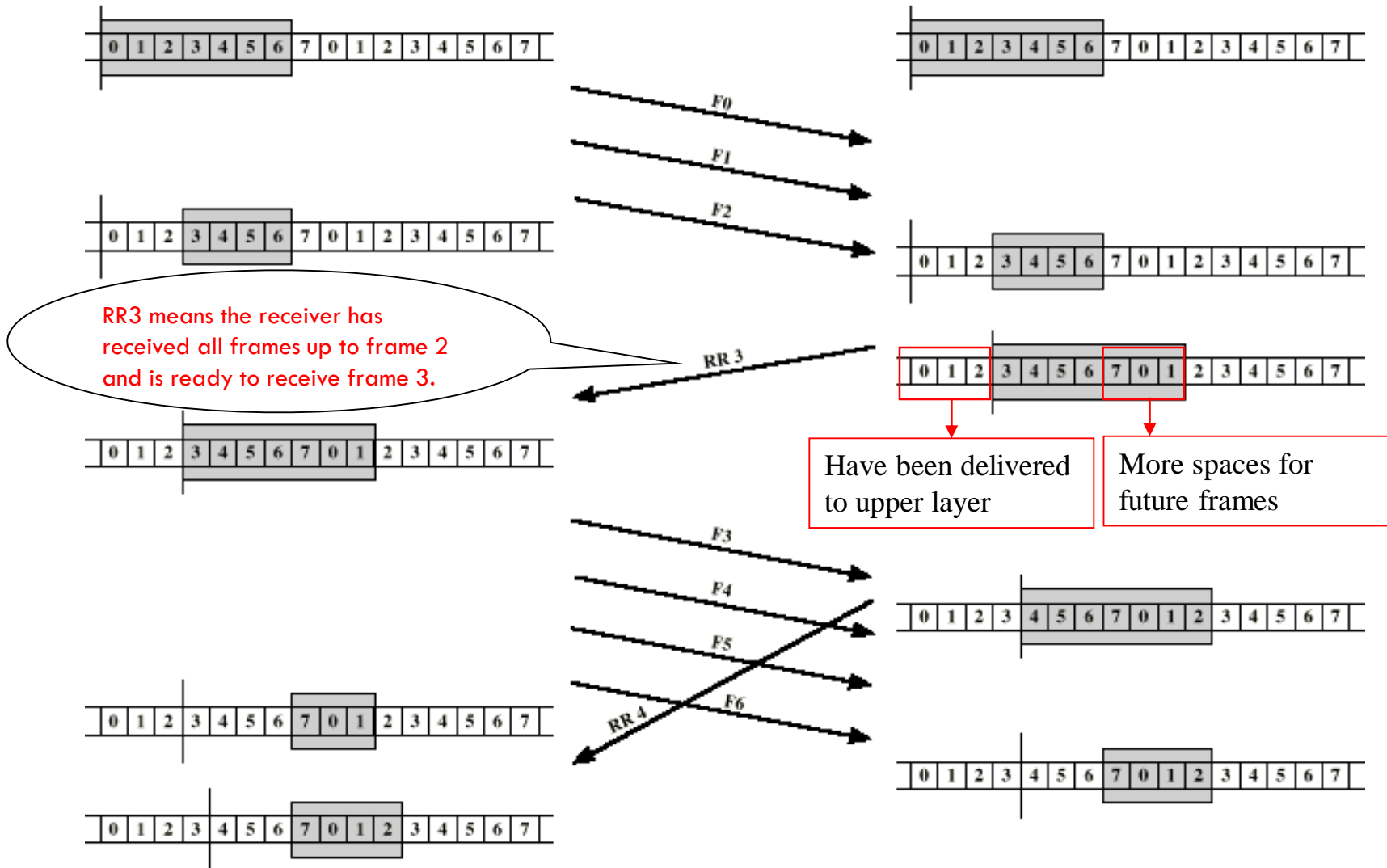
A sliding window with a 2-bit sequence, of size 1

Example Sliding-Window

39

Source System A

Destination System B



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.

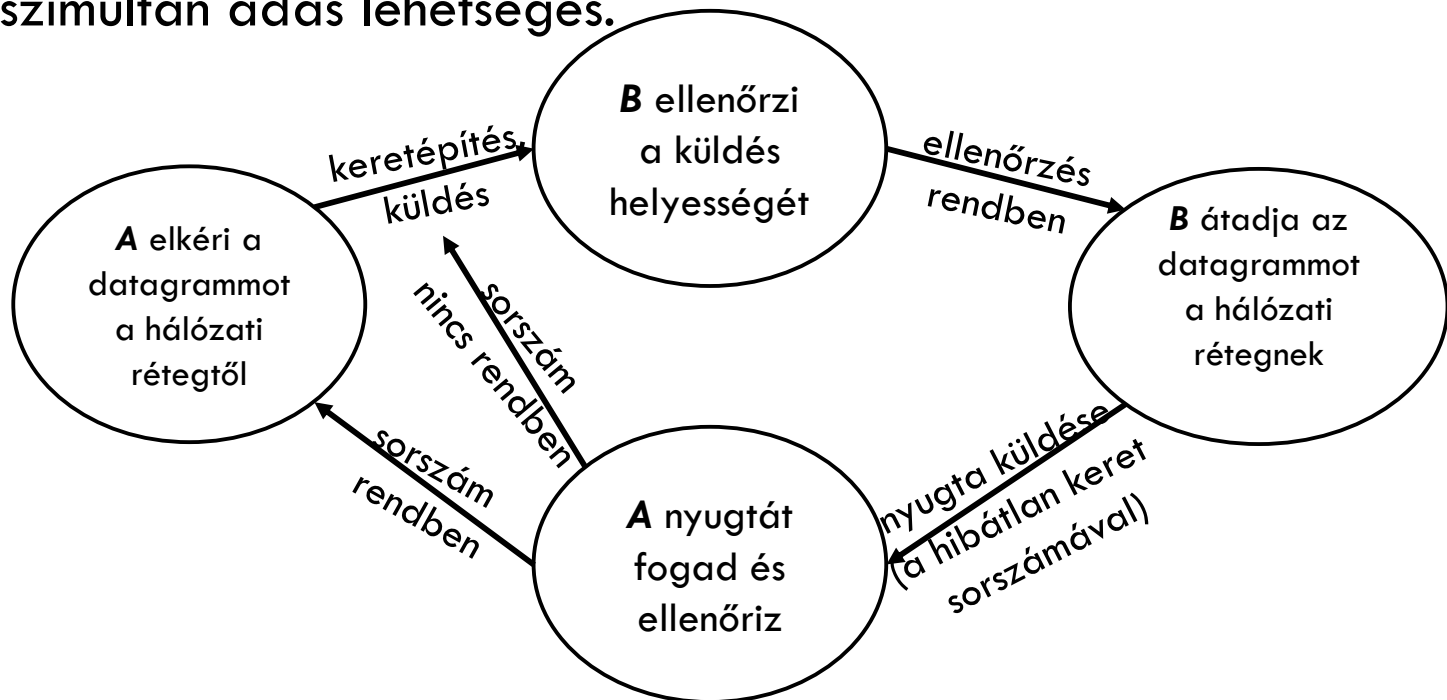
JELLEMZŐK (GYAKORLATI ALKALMAZÁS ESETÉN)

- 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*

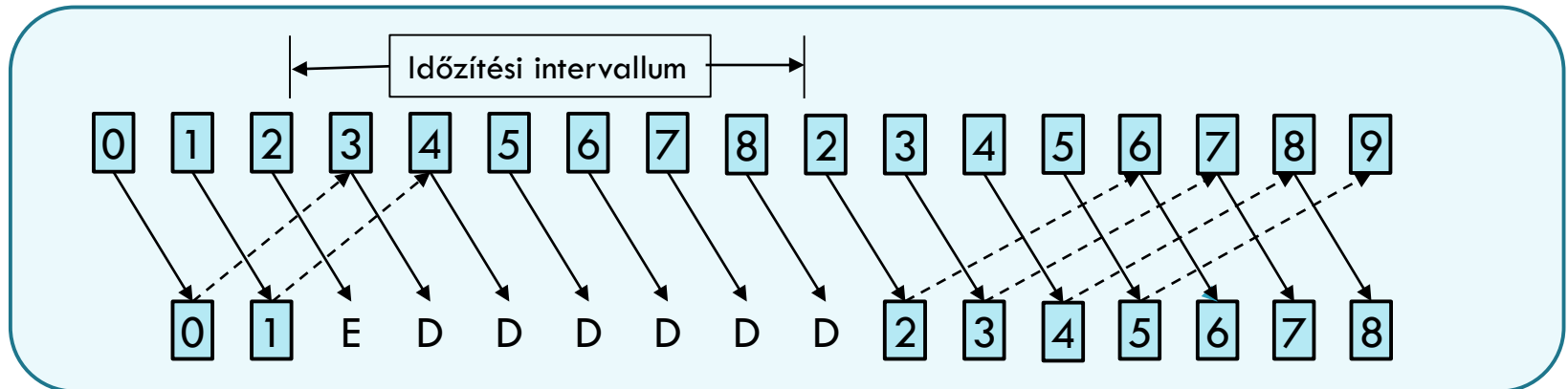
„visszalépés N-nel” stratégia

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.



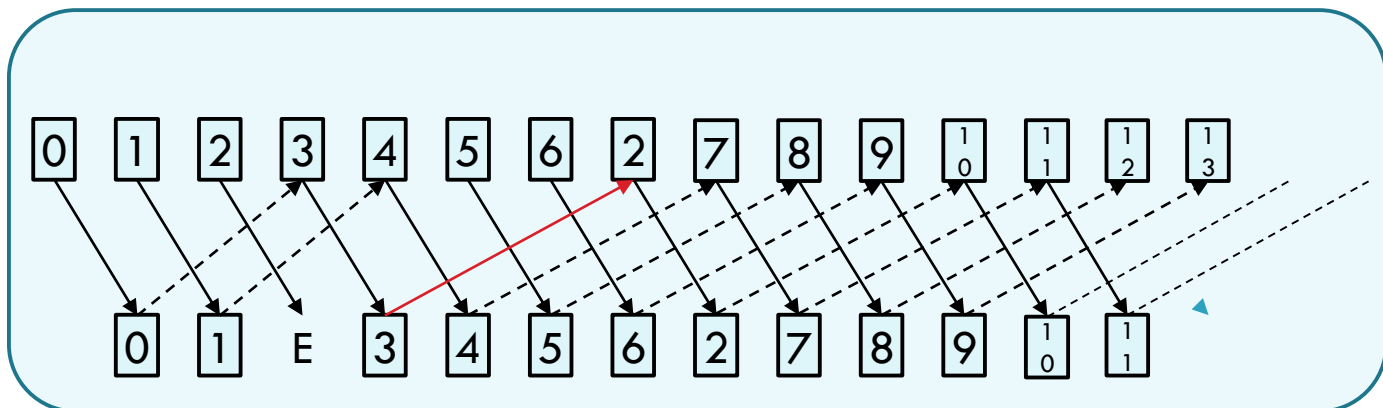
„szelektív ismétlés” stratégia

Stratégia lényege

- A hibás kereteket eldobja, de a jó kereteket a hibás után puffereleli.
- 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 ?

48

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

MAC alréteg

49

- 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

50

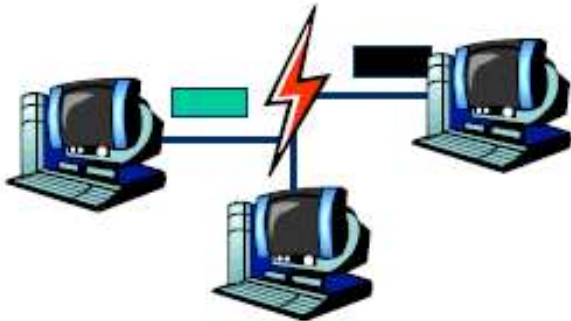
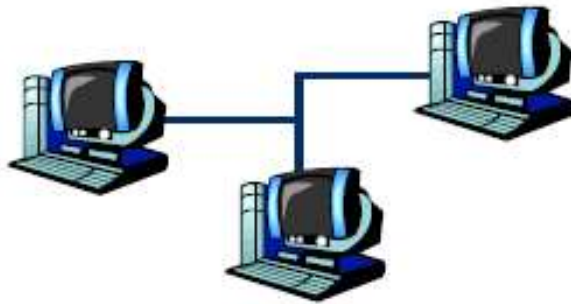
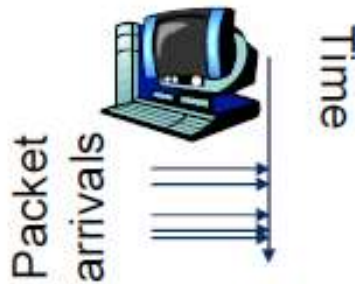
Frekvenciaosztásos nyalábolás

- 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 $\lambda \Delta t$, ahol λ az érkezési folyamat rátája.

2. Egyetlen csatorna feltételezés

- ▣ Minden állomás egyenrangú.
- ▣ Minden kommunikáció egyazon csatornán zajlik.
- ▣ Minden állomás tud ezen küldeni és fogadni csomagot.

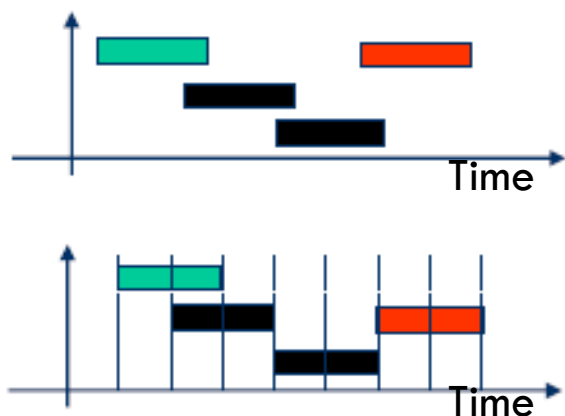
3. Ütközés feltételezés

- ▣ 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

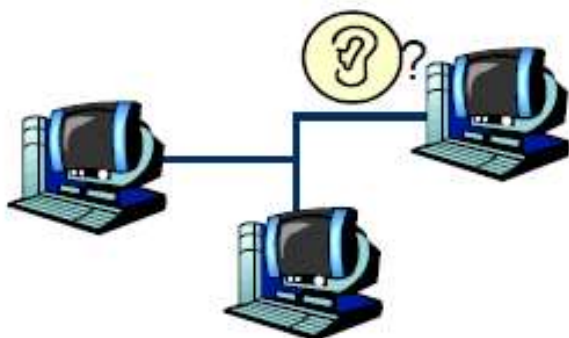
Kétféle időmodellt különböztetünk meg:

- 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 elkezdnek 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

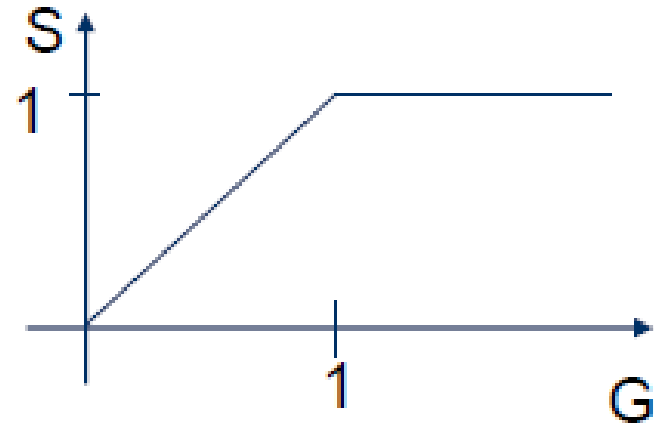
□ Fairség [Fairness]

- ▣ 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

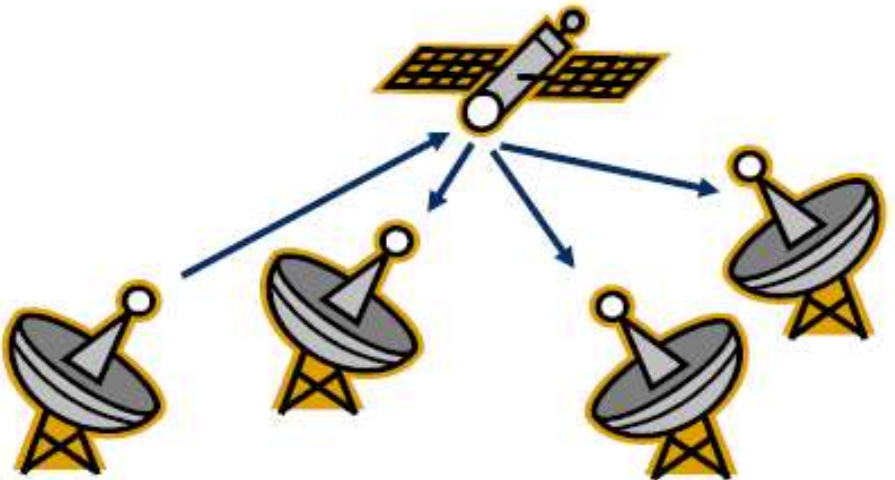
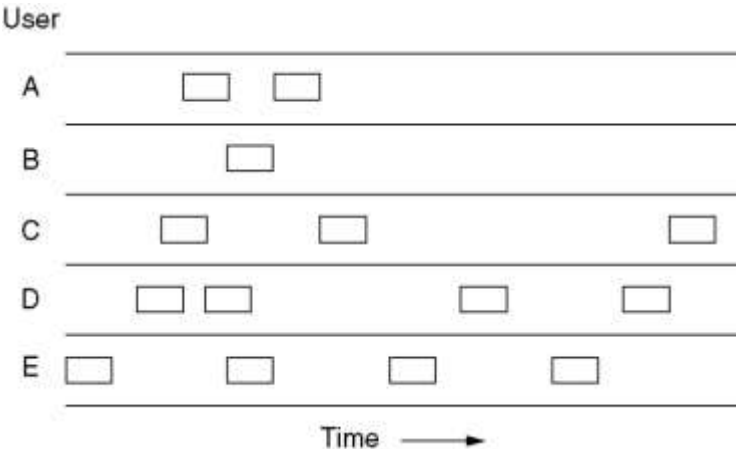


□ Ideális esetben

- Ha $G < 1$, $S = G$
- Ha $G \geq 1$, $S = 1$
- Ahol egy csomag kiküldése egy időegységet vesz igénybe.

[illegible]

- Alacsony költségű, nagyon egyszerű megoldás



ALOHA

56

- ❑ 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.
 - Kevés küldő esetére készült

Teljesítmény elemzés -Poisson Folyam

- A „**véletlen érkezések**” egyik ünnepelt modellje a sorban-állás elméletben a Poisson folyamat.
- 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

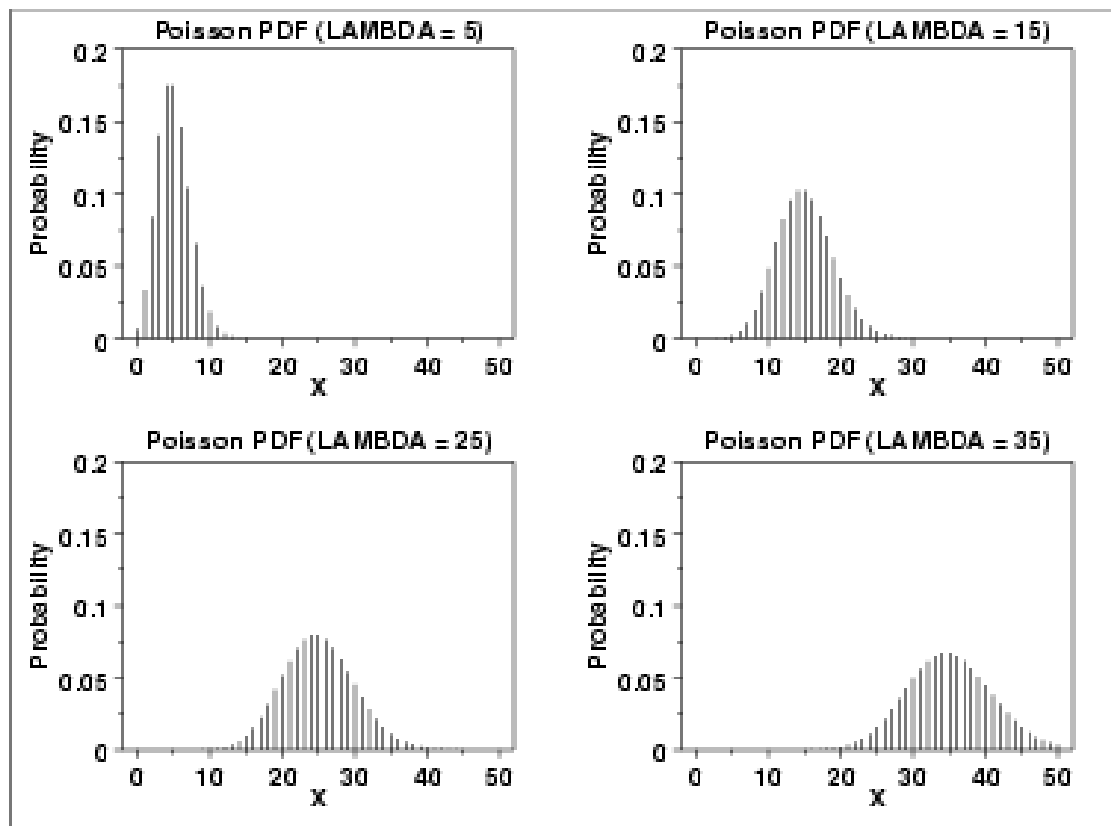
Annak a valószínűsége, hogy k érkezés történik egy t hosszú intervallum során:

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

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

Poisson Eloszlás példák

59



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

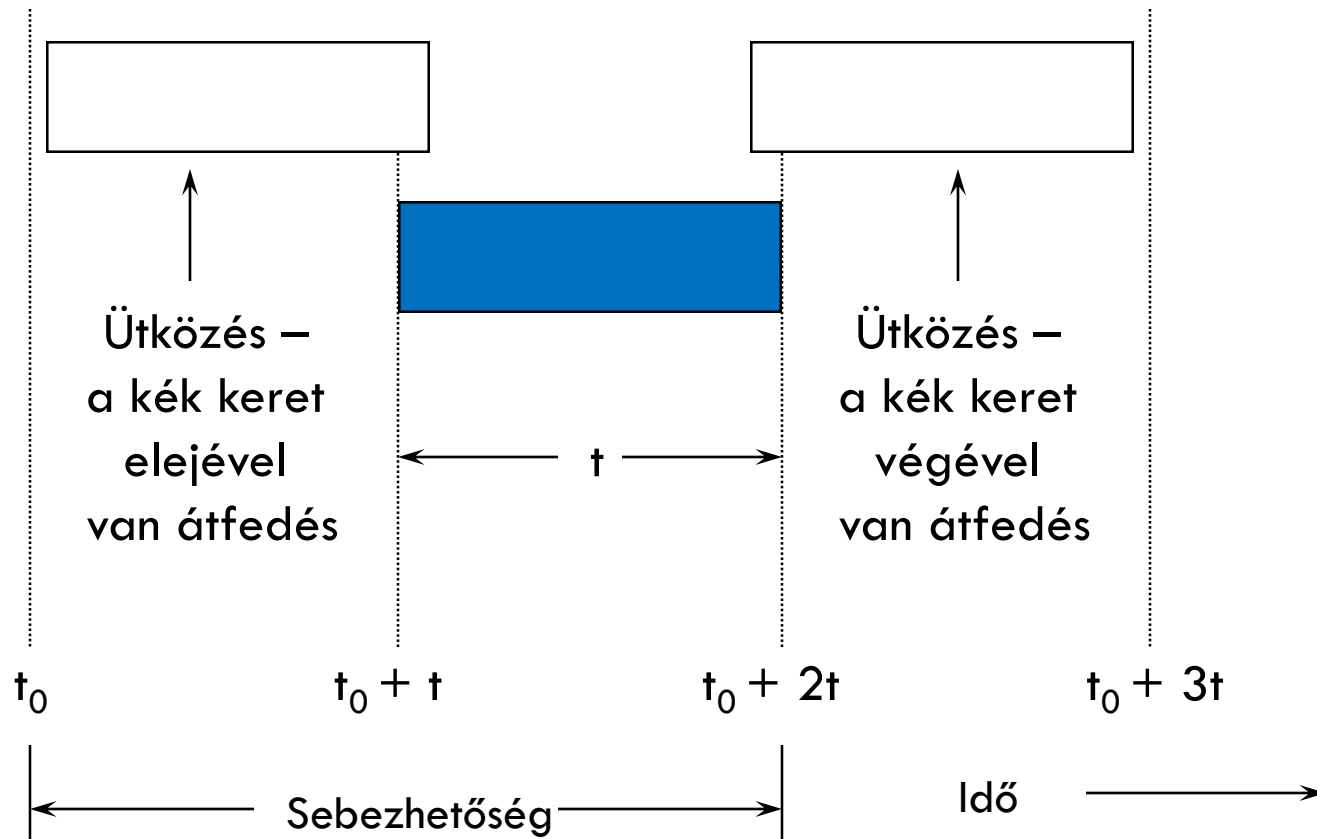
- Mivel S jelöli a „jó” átviteleket egy keret idő alatt és G jelöli az összes átviteli kísérletet egy keret idő alatt, így a következő összefüggést írhatjuk:

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

- 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

62



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)

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

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

ALOHA vizsgálata

64

- $S(G) = Ge^{-2G}$ 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

65

□ A TDMA

□ A v

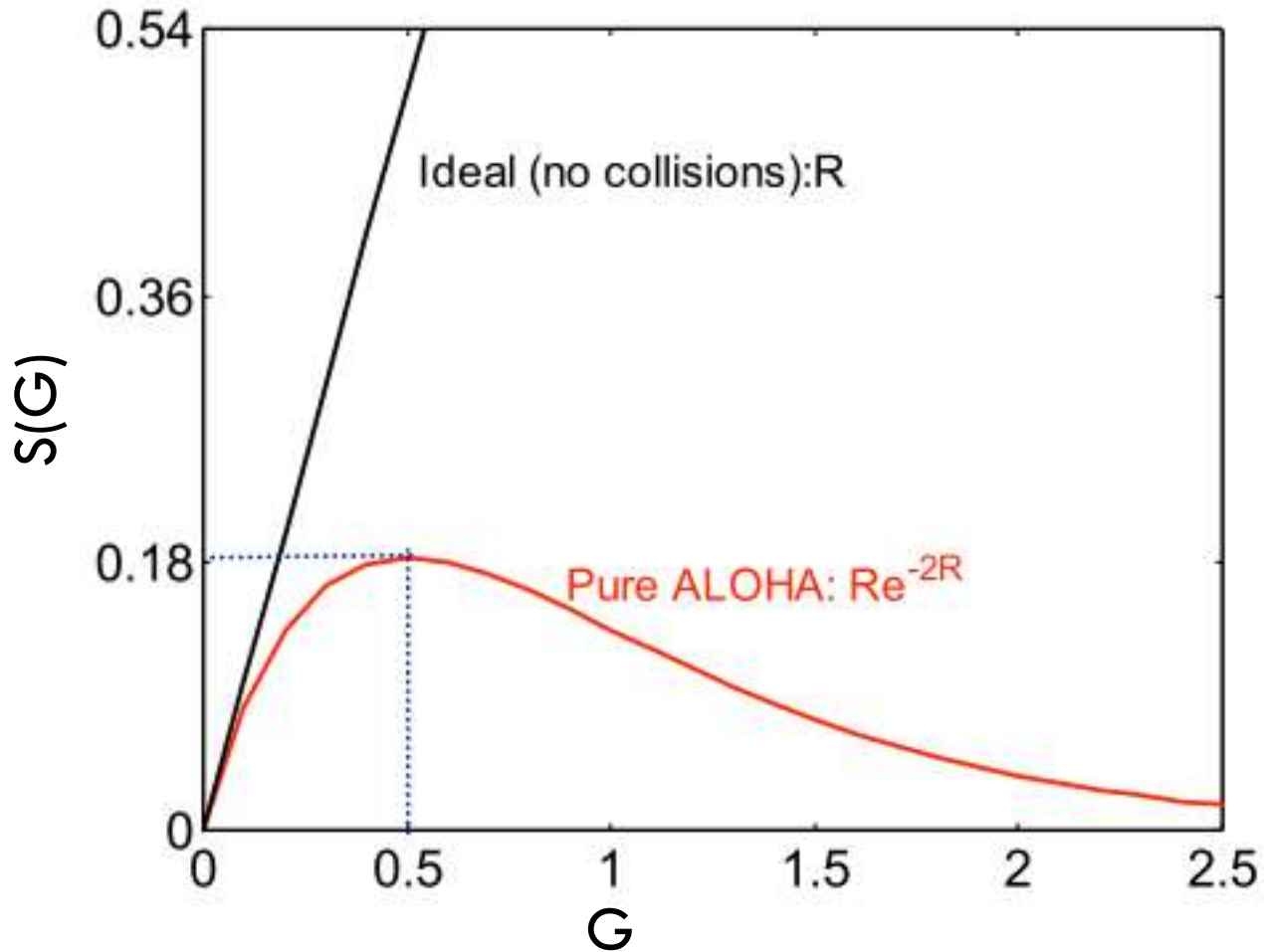
□ Az A

□ Sol

□ De

Sender

Sender

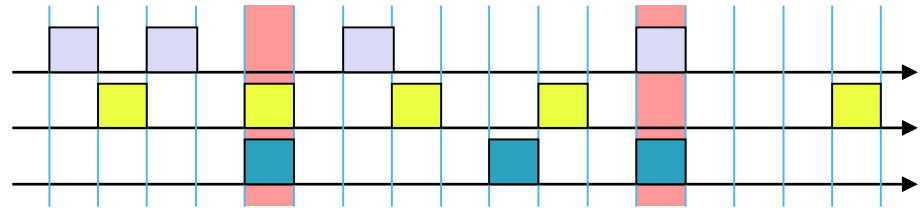


□ Maximum system capacity is 18% of the

Réselt ALOHA

66

- A csatornát azonos időrésekre bontjuk, melyek hossza pont egy keret átviteléhez szükséges idő.
- Átvitel csak az időrészek 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

68

□ Protokoll

□ Ugrás

■ R

□ Cs

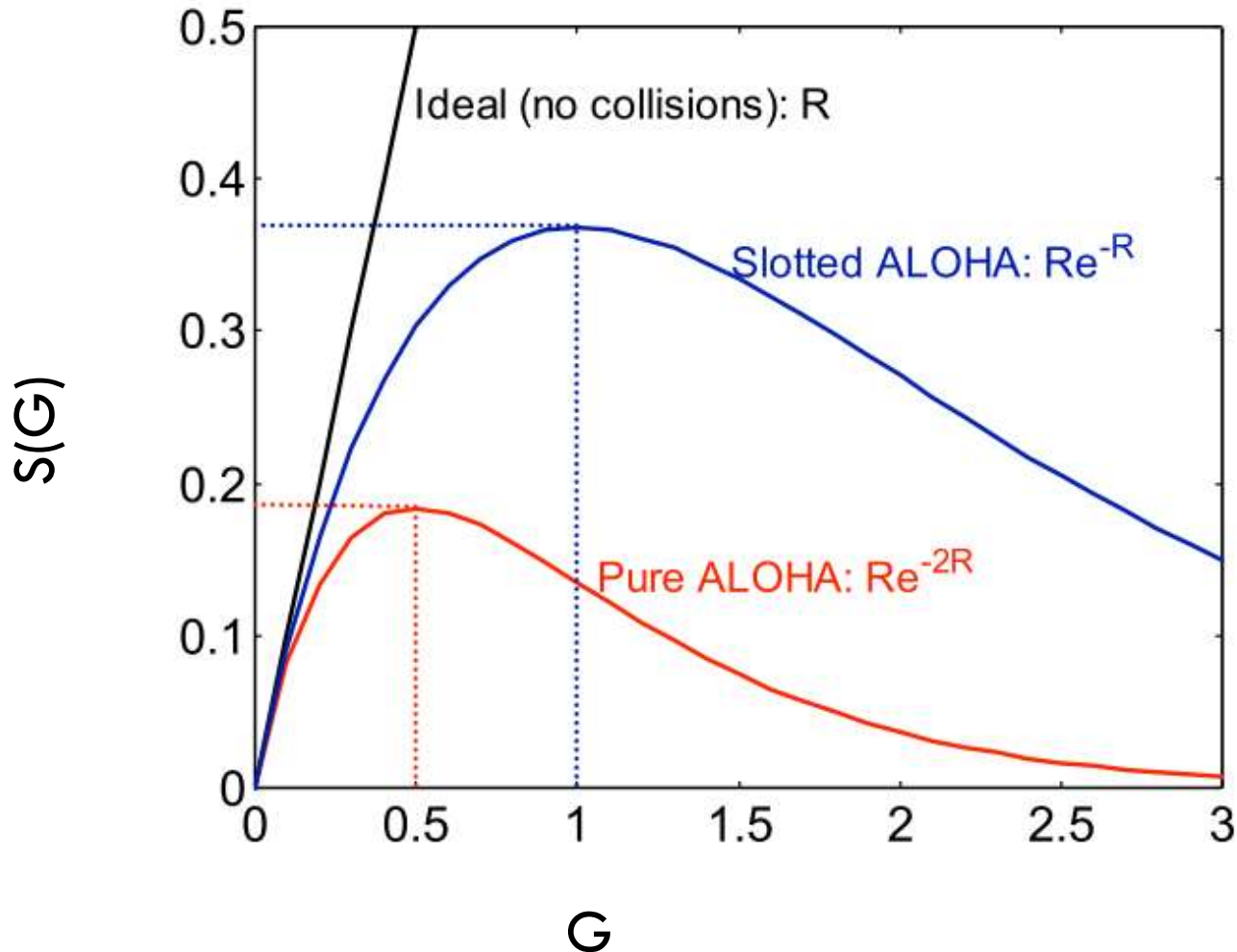
□ Azaz

nem

□ 37

□ Az

ke



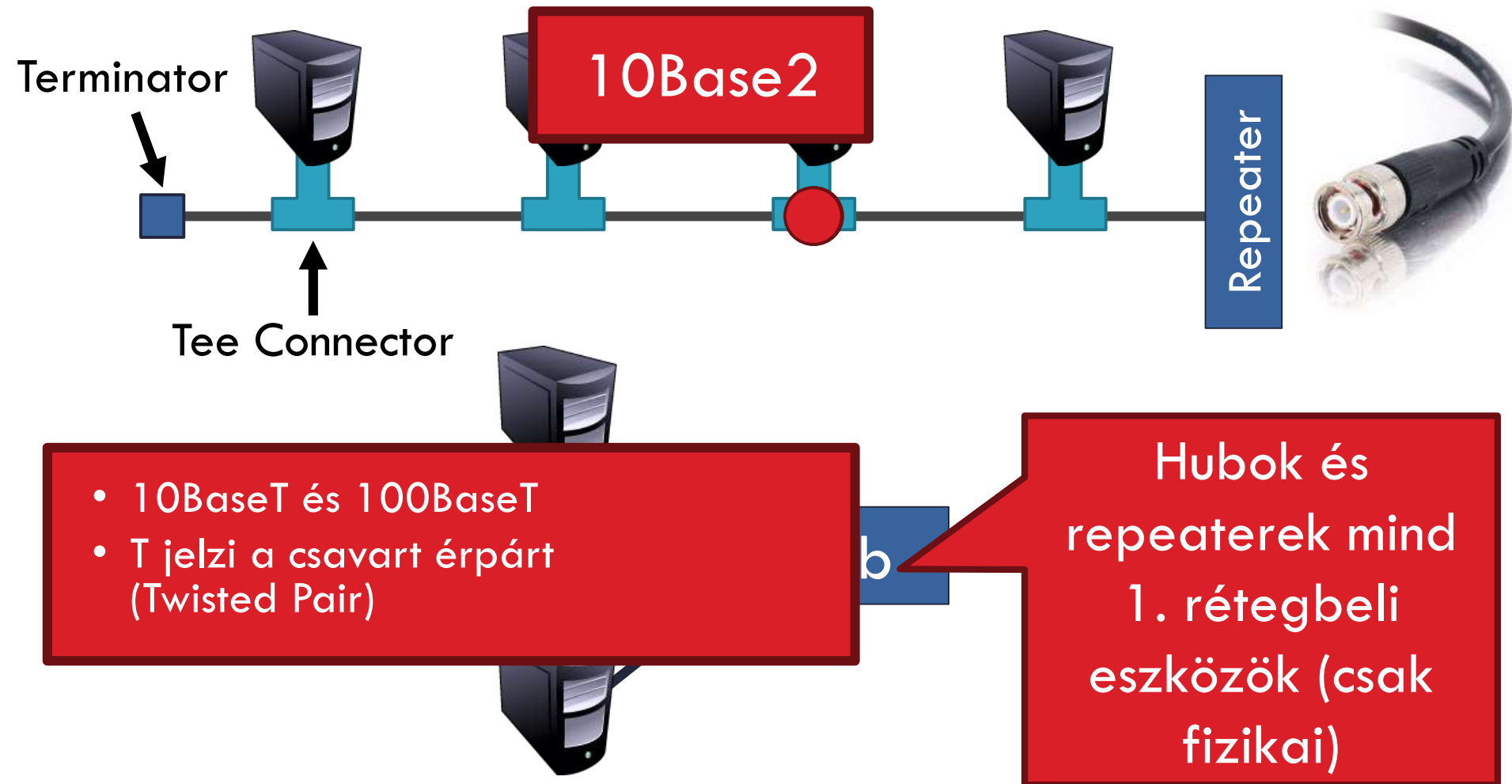
általán

órával

Adatszóró (Broadcast) Ethernet

69

- Eredetileg az Ethernet egy adatszóró technológia volt



Vivőjel érzékelés

Carrier Sense Multiple Access (CSMA)

- További feltételezés

- ▣ Minden állomás képes beleszállni a csatornába és így el tudja dönteni, hogy azt más állomás használja-e átvitelre

1-perzisztens CSMA protokoll

71

- 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

72

- 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 újratekzi a keret leadását.

Tulajdonságok

- Jobb teljesítményt mutat, mint az 1-perzisztens CSMA protokoll. (*intuitív*)

p-perzisztens CSMA protokoll

73

- Vivőjel érzékelés van, azaz minden állomás belehallgathat a csatornába.
- Diszkrét időmodellt használ a protokoll

Algoritmus

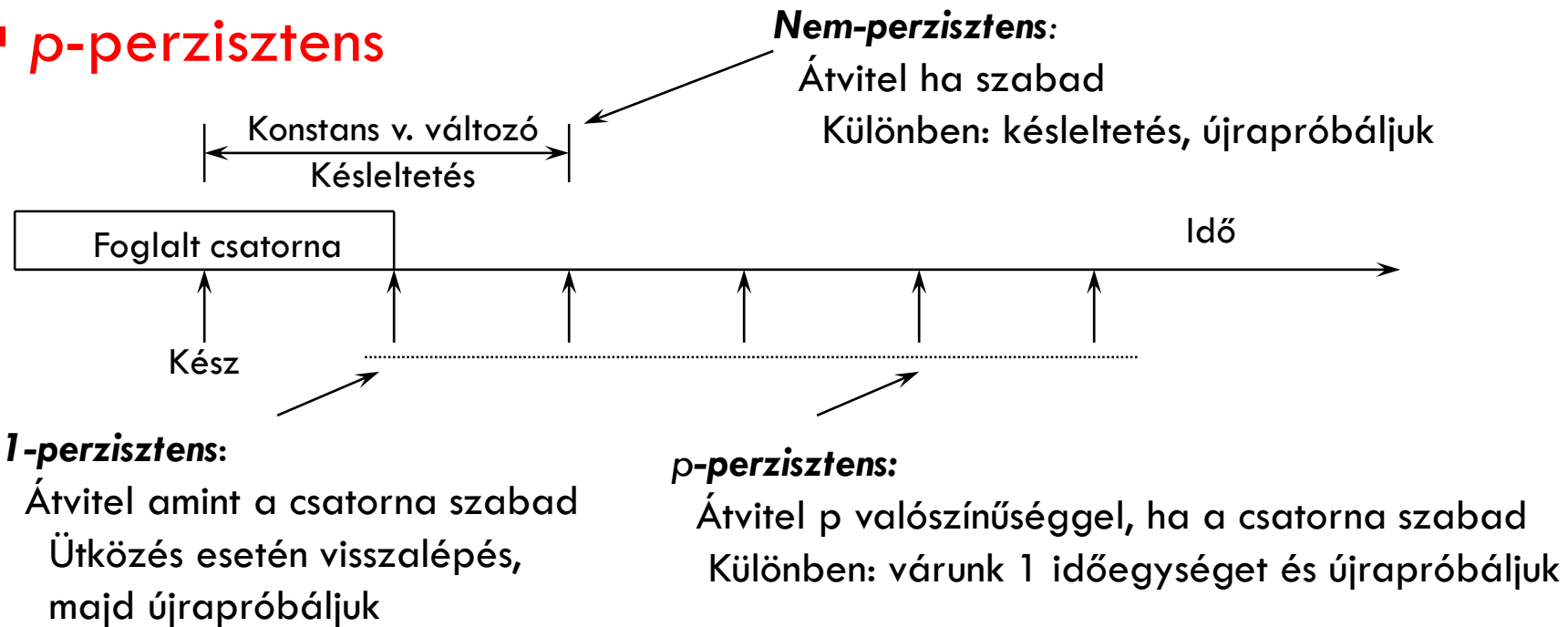
- Adás kész állapotban az állomás belehallgat a csatornába:
 - a) Ha foglalt, akkor vár a következő időrésig, majd megismétli az algoritmust.
 - b) Ha szabad, akkor p valószínűséggel küld, illetve $1-p$ valószínűséggel visszalép a szándékától a következő időrésig. Várakozás esetén a következő időrésben 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

74

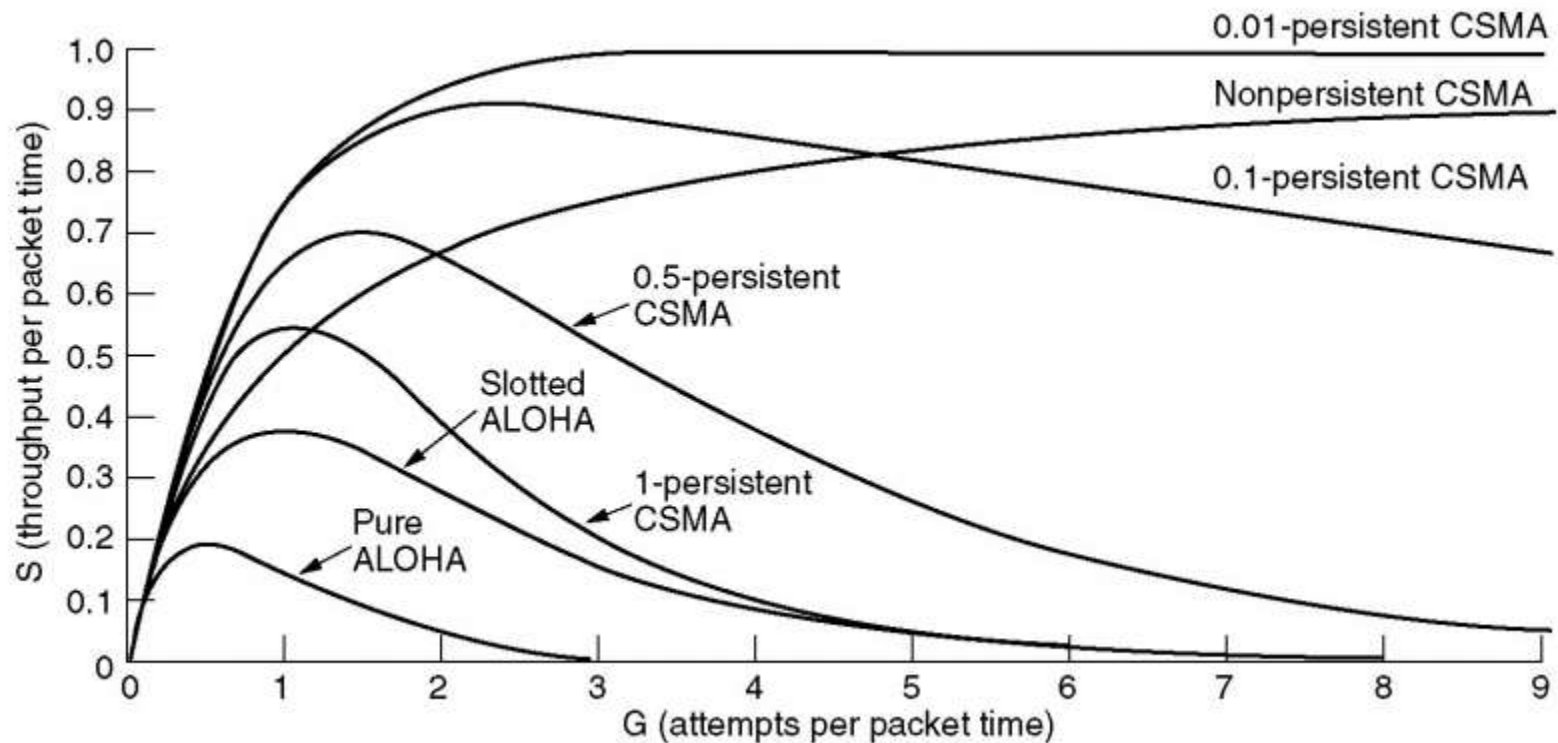
- Nem-perzisztens
- 1-perzisztens
- p -perzisztens

CSMA perzisztencia



CSMA és ALOHA protokollok összehasonlítása

75



Köszönöm a figyelmet!