

Számítógépes Hálózatok

4. Előadás: Adatkapcsolati réteg II.

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

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.

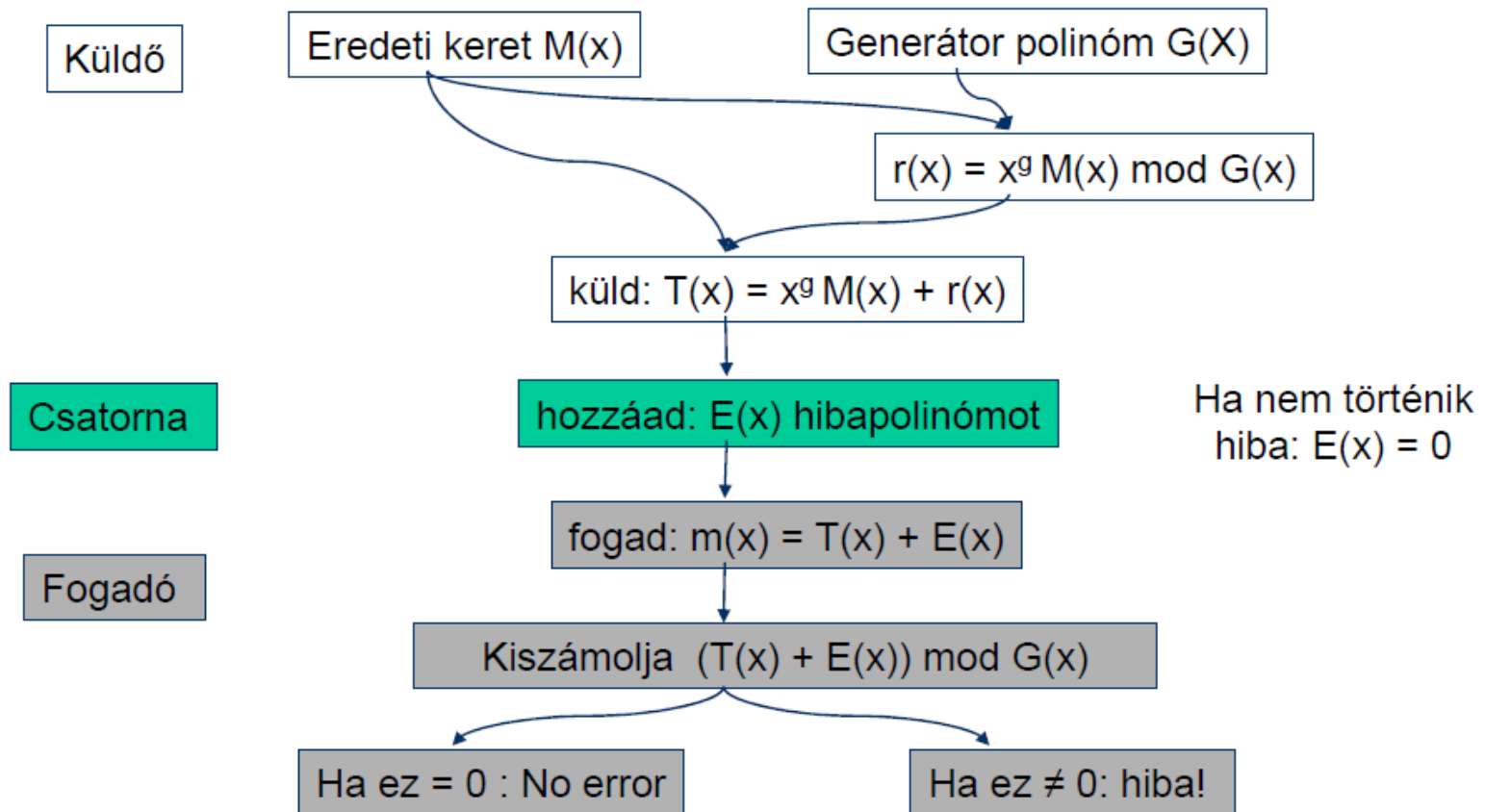
Algoritmus

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

5

□ Forrás: Dr. Lukovszki Tamás fóliái

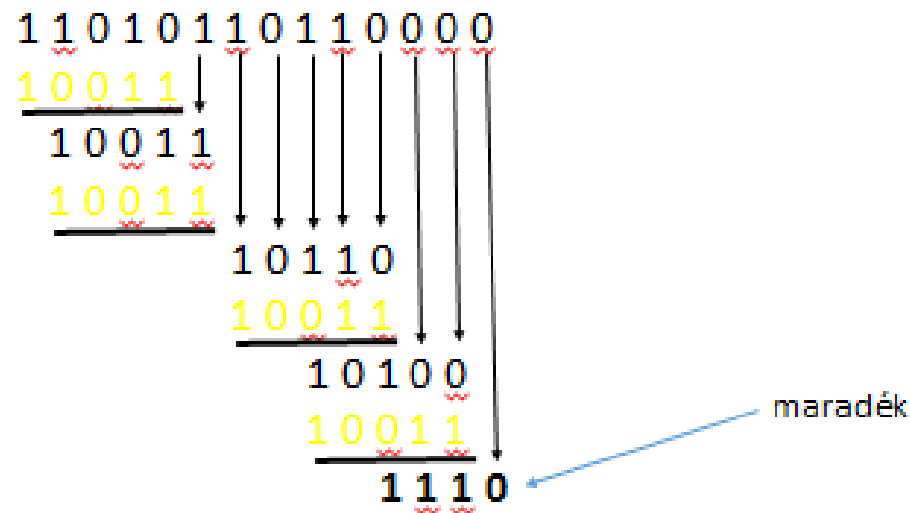


Példa CRC számításra

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.

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

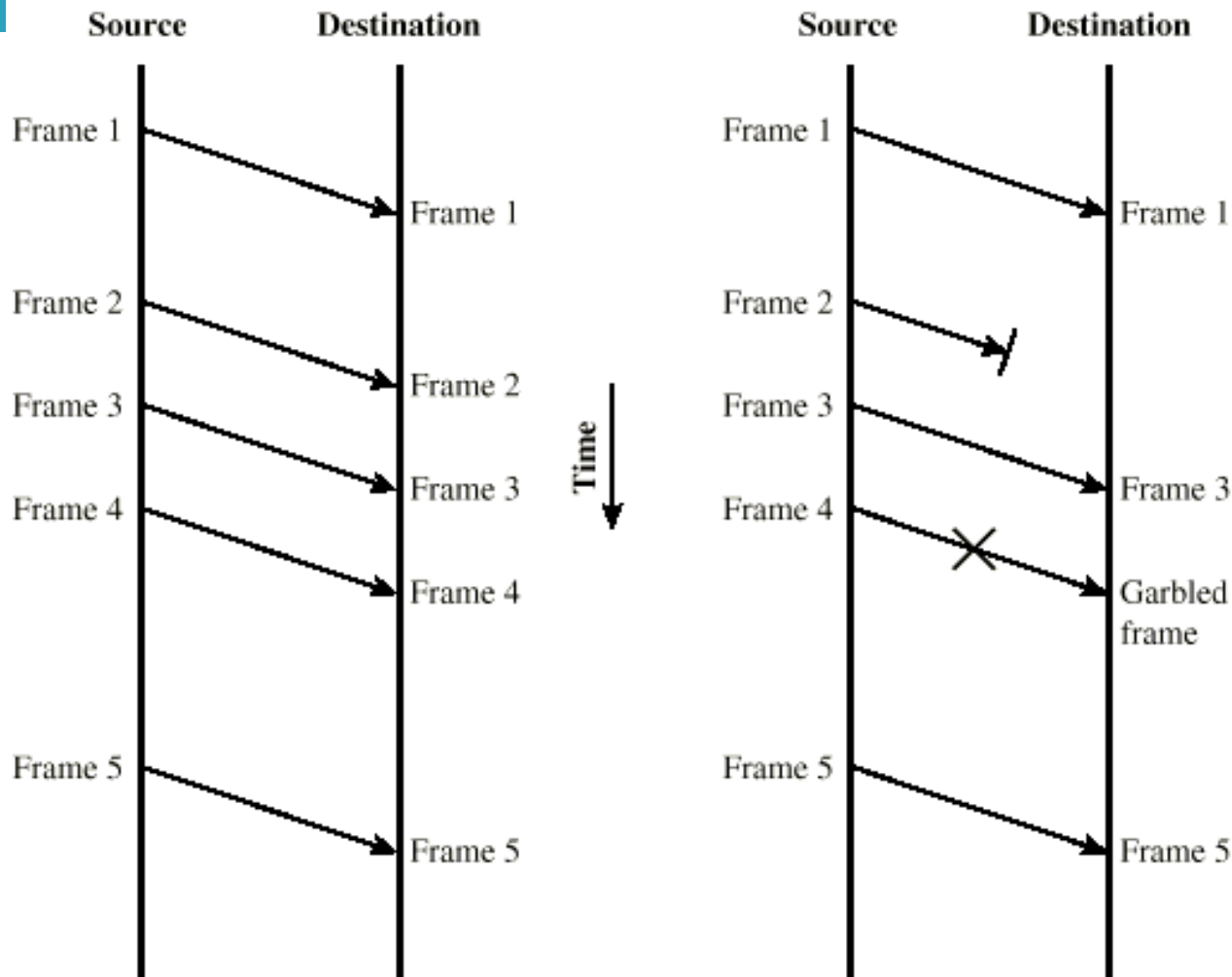
- mind az adó, mind a vevő hálózati rétegei mindig készen állnak;
- a feldolgozási időktől eltekintünk;
- végtelen puffer-területet feltételezünk;
- Az adatkapcsolati rétegek közötti kommunikációs csatorna sosem rontja vagy veszíti el a kereteket;

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

13



(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 puffereles 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.

Megáll-és-vár

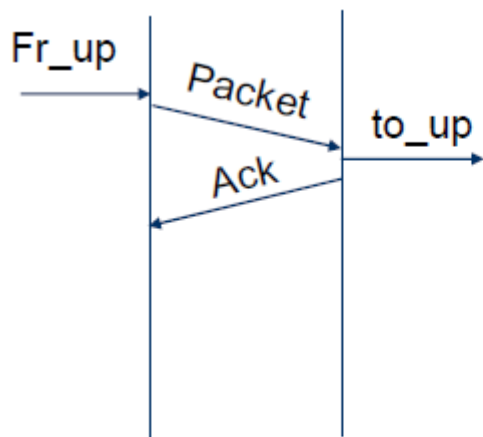
16

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

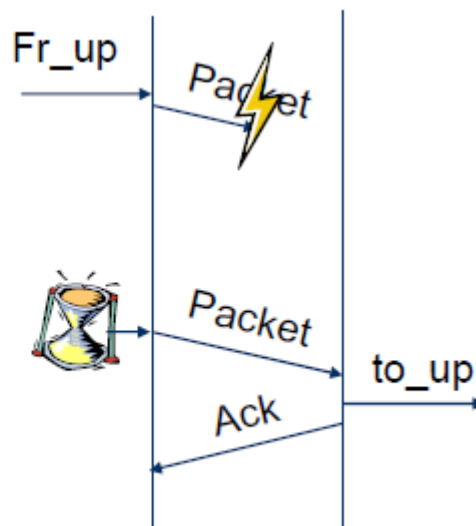
Küldhetnénk egymás után
folyamatosan???

Mi is a probléma?

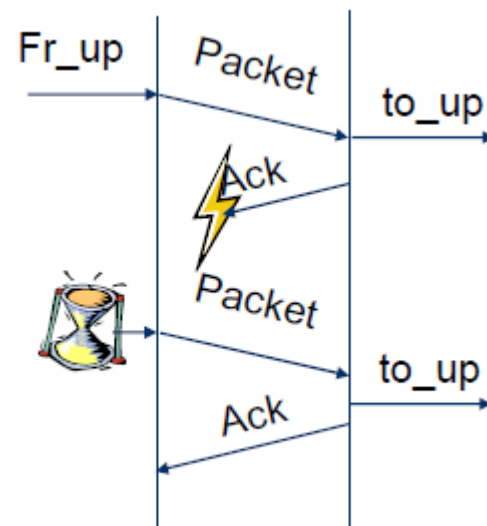
Általában



Csomagvesztés esetén

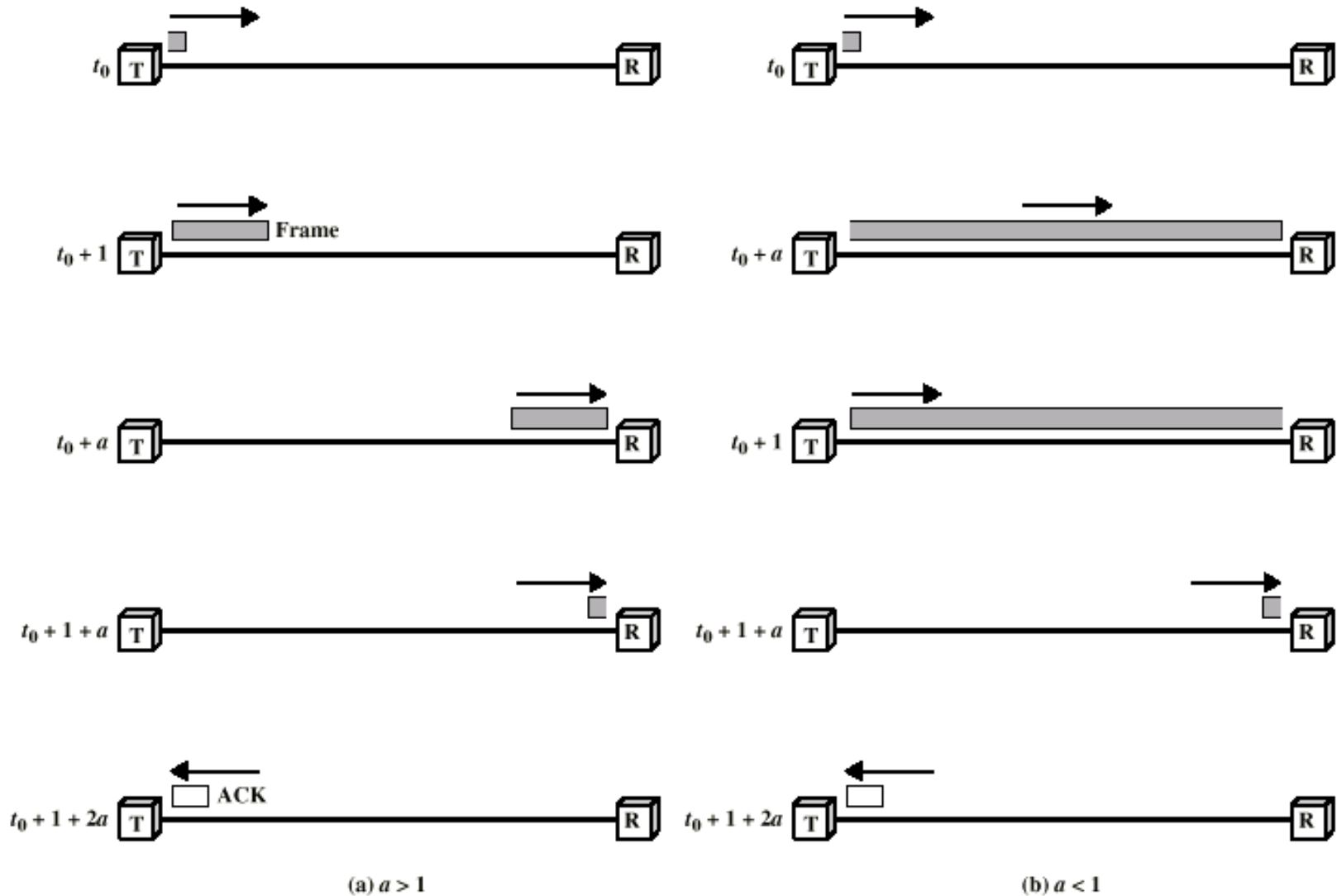


ACK vesztés esetén



Csatorna kihasználtság

18



Alternáló-bit protokoll (ABP)

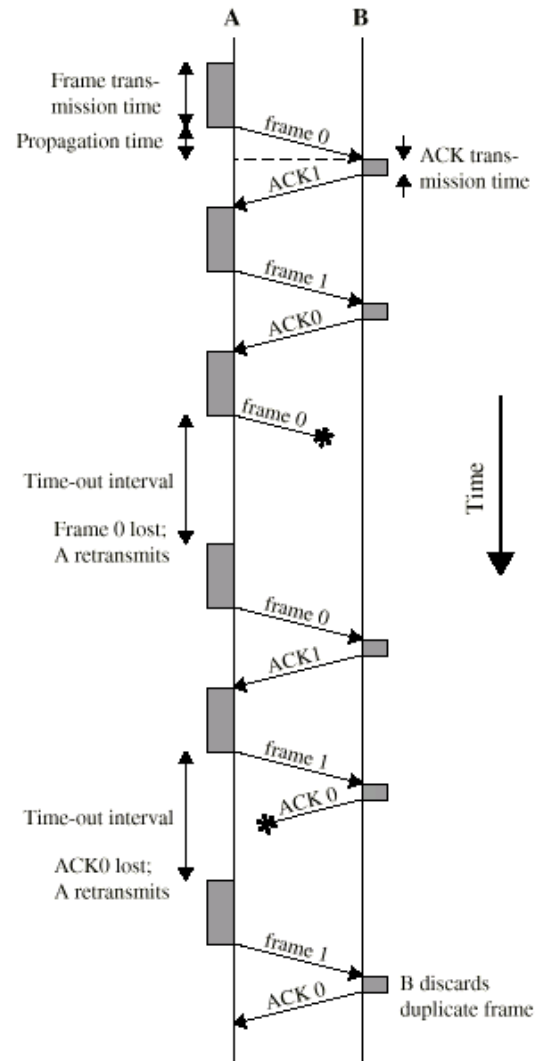
- ❑ **Megoldás:** sorszámok használata
- ❑ Mennyi sorszámra lesz szükség? $\{0,1\}$ elegendő

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.

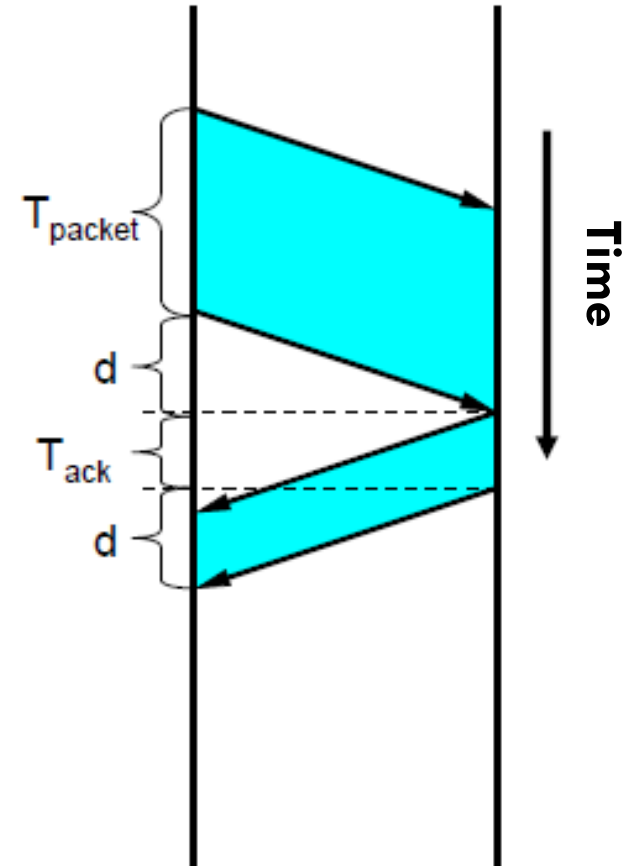
ABP

20



ABP – Csatorna kihasználtság

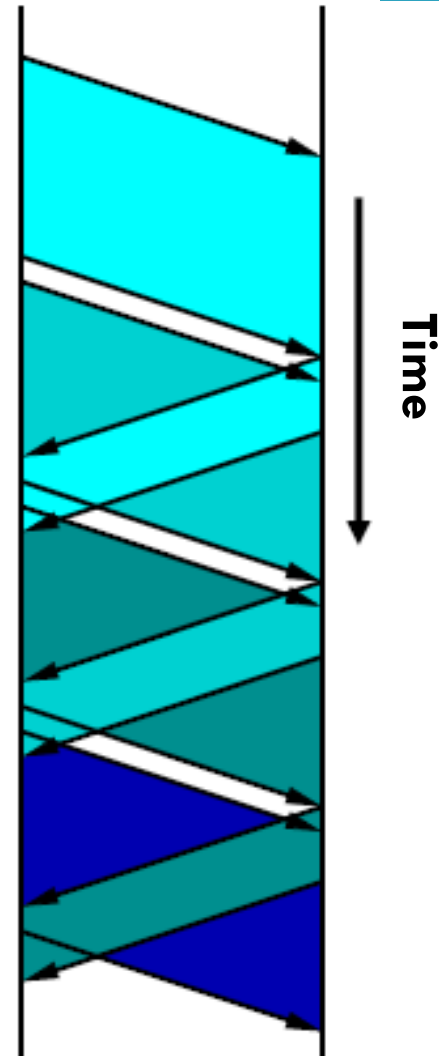
- Kihasználtság (η) a következő két elem aránya
 - ▣ A csomag elküldéséhez szükséges idő (T_{packet})
 - ▣ Az idő, ami a következő keret küldéséig eltelik
 - Az ábrán: ($T_{\text{packet}} + d + T_{\text{ack}} + d$)
- ABP esetén:
 - ▣ $\eta = T_{\text{packet}} / (T_{\text{packet}} + d + T_{\text{ack}}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

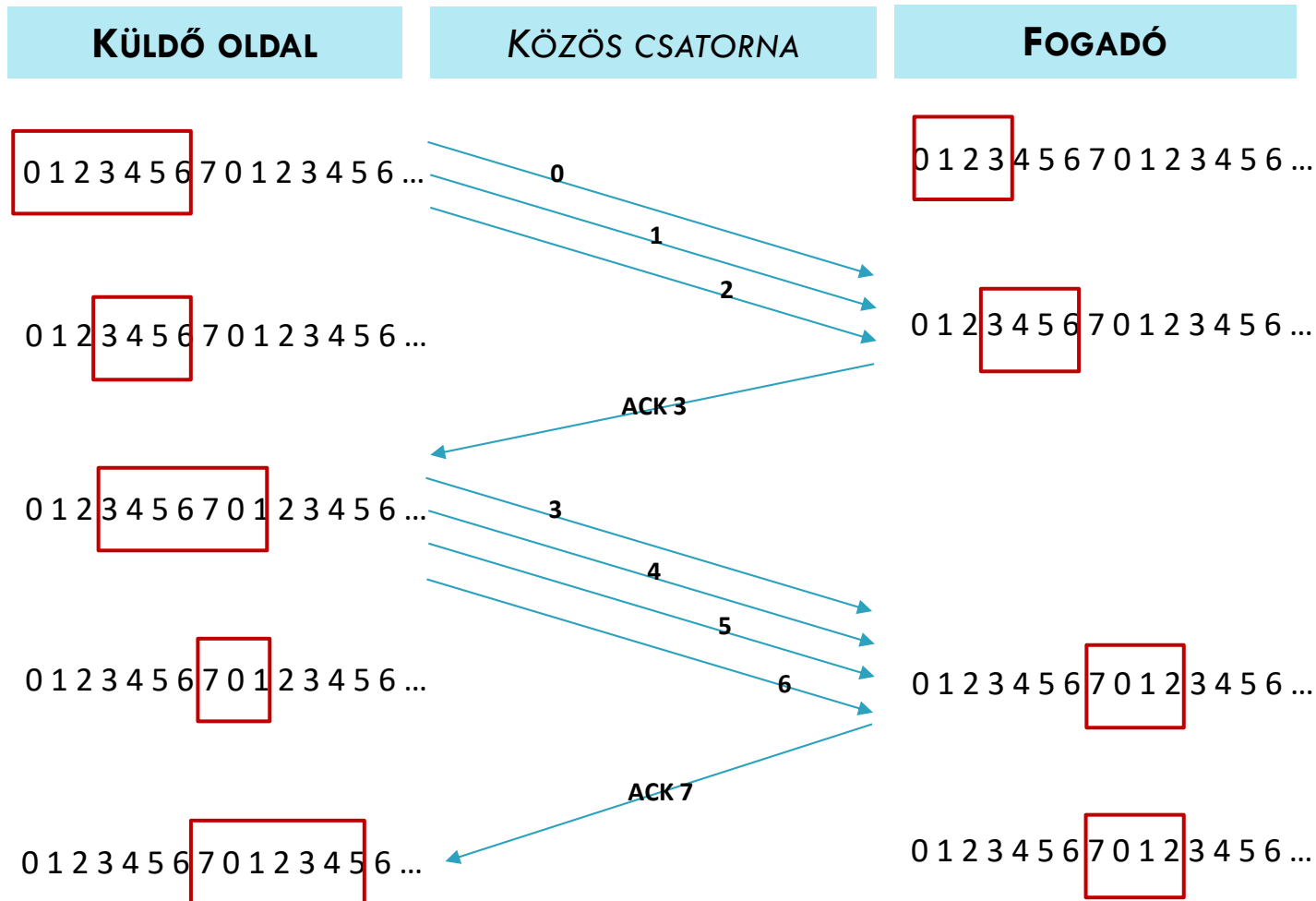
ALAPOK (ÁLTALÁNOS)

- 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.

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.

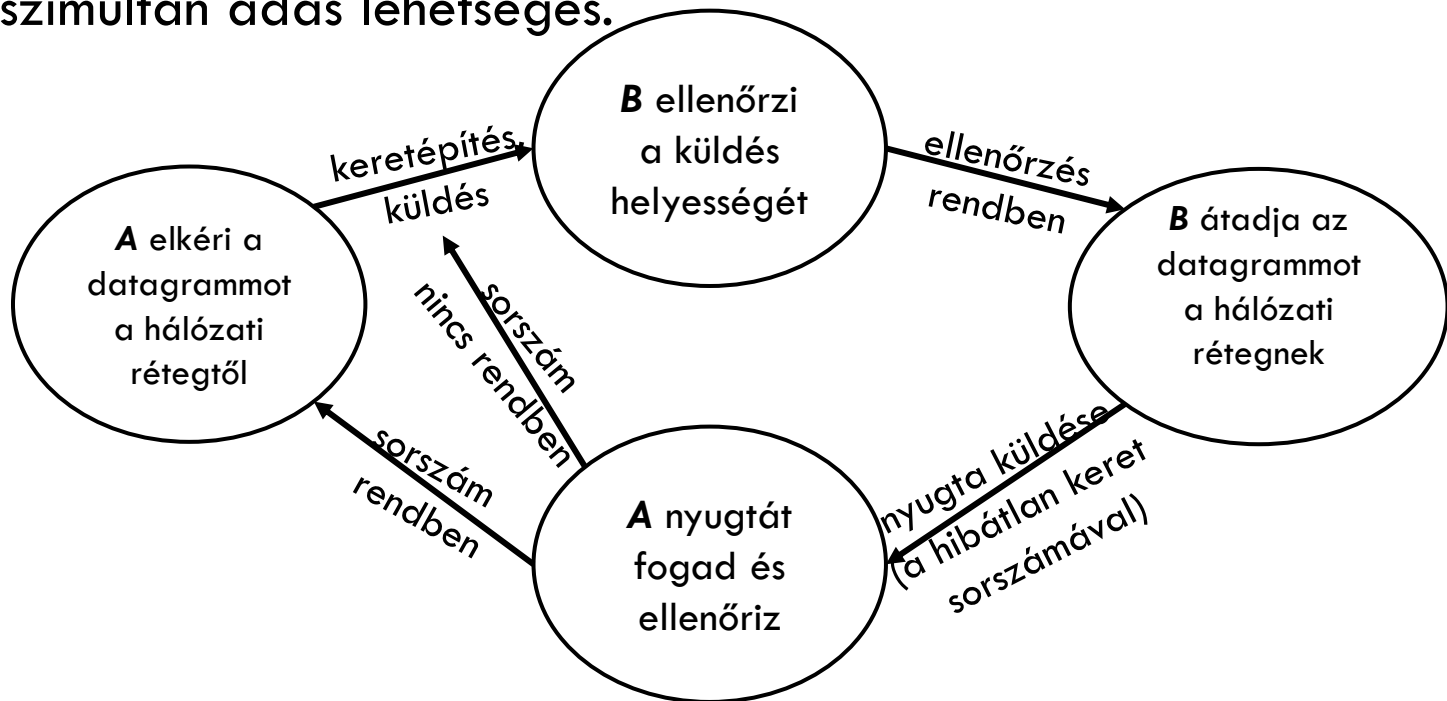
JELLEMZŐK (GYAKORLATI ALKALMAZÁS ESETÉN)

- 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*);

Egy bites 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*

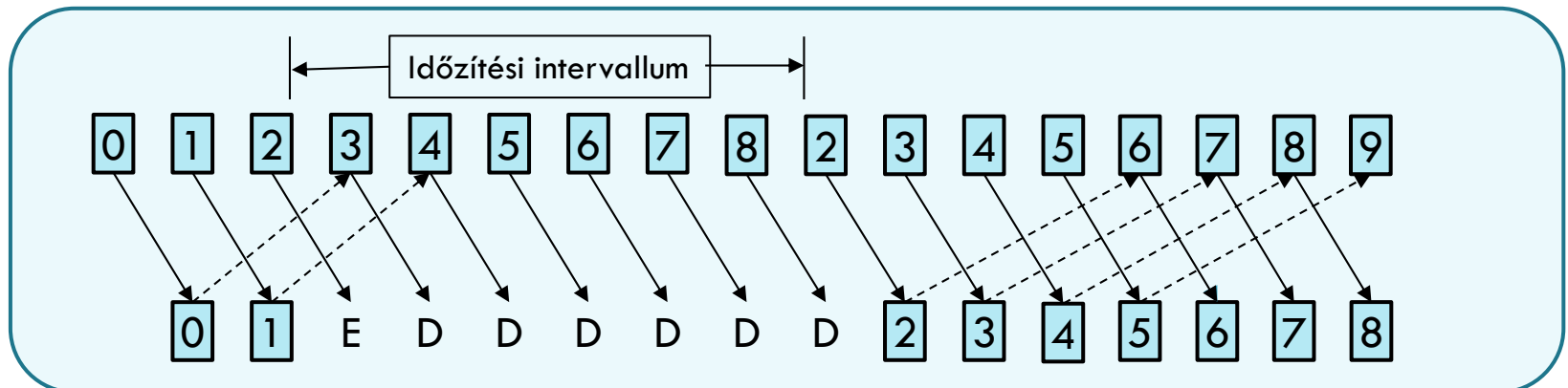
„visszalépés N-nel” stratégia

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.



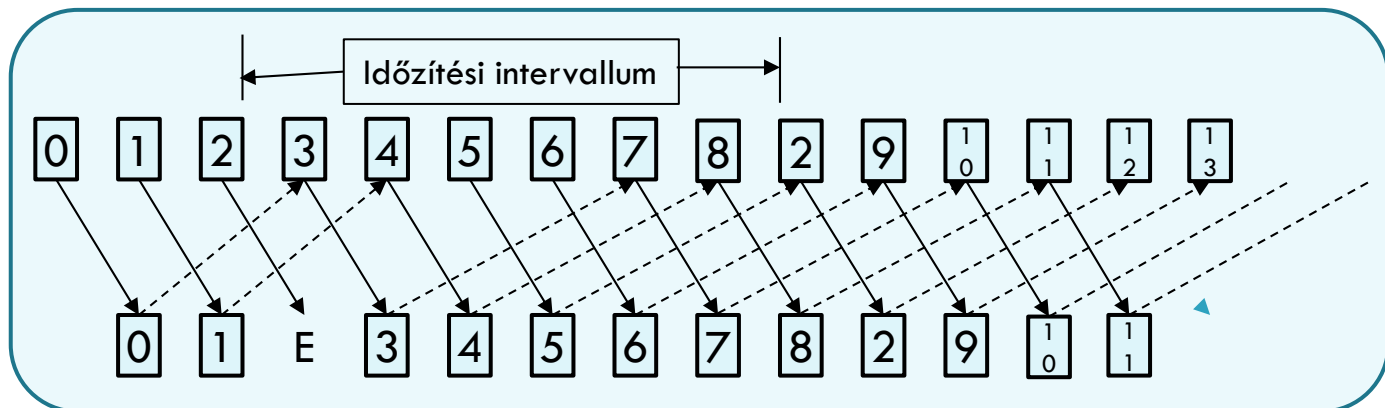
„szelektív ismétlés” stratégia

Stratégia lényege

- A hibás kereteket eldobja, de a jó kereteket a hibás után pufferezi.
- 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.



Példák az adatkapcsolati réteg protokolljaira

HDLC 1/4

31

ÁTTEKINTÉS

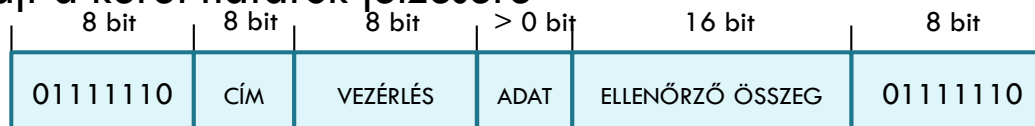
- IBM mainframe világában használták először az SDLC protokoll;
 - ▣ ANSI szabvány változat ADCCP
 - ▣ ISO szabvány változat HDLC
 - ▣ Később a CCITT adoptálta a LAP és a LAPB protokolljához.
- A különböző szabványok közös elveken nyugszanak
 - ▣ Bitorientáltság (bitbeszúrás alkalmazása)
 - ▣ Apróbb sajátosságok

HDLC 2/4

32

ÁLTALÁNOS KERETFELÉPÍTÉS

- cím mező
 - ▣ több vonallal rendelkező terminálok esetén van jelentősége,
 - ▣ pont-pont kapcsolatnál parancsok és válaszok megkülönböztetésére használják
- vezérlés mező
 - ▣ sorszámozás, nyugtázás és egyéb feladatok ellátására
- adat mező
 - ▣ tetszőleges hosszú adat lehet
- ellenőrző összeg mező
 - ▣ CRC kontrollösszeg a CRC-CCITT generátor polinom felhasználásával
- FLAG bájt a keret határok jelzésére



HDLC 3/4

33

- 3 bites csúszó-ablak protokoll használ a sorszámozáshoz
- Három típusú keretet használ:
 - ▣ információs
 - ▣ felügyelő

Típusok

- 0. típus – nyugtakeret (RECEIVE READY);
- 1. típus – negatív nyugtakeret (REJECT);
- 2. típus – VÉTELRE NEM KÉSZ, amely nyugtáz minden keretet a Következőig ((RECEIVE NOT READY))
- 3. típus – SZELEKTÍV ELUTASÍTÁS, amely egy adott keret újraküldésére szólít fel (SELECTIVE REJECT)
- ▣ Számozatlan
 - típus 5 bites adat

HDLC 4/4

34

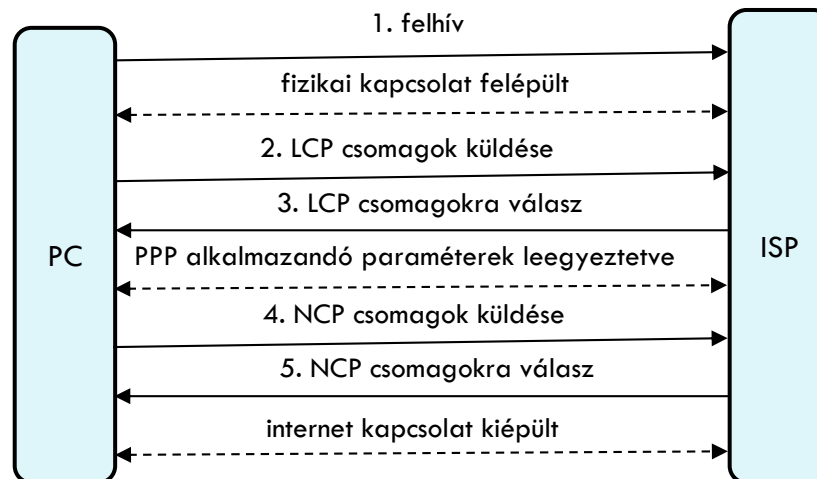
LEGELTERJEDTEBB PARANCSONK SZÁMOZATLAN KERETEK ESETÉN

- ❑ DISC – a csatlakozás bontására vonatkozó szándék bejelentésére
- ❑ SNRM – visszacsatlakozó állomás bejelentse a jelenlétét (*aszinkron*)
- ❑ SABM – alaphelyzetbe állítja a vonalat (*egyenrangúság*)
- ❑ Kiterjesztett keretformátum (7 bites sorszám) engedélyezésekor: SAMBE, SNRME.
- ❑ FRMR – keretelutasítás illegális keretek jelzésére

PPP 1/2

35

- RFC 1661, 1662, 1663
- **Három dolgot biztosít**
 1. Keretezési módszert (egyértelmű kerethatárok).
 2. Kapcsolatvezérlő protokollt a vonalak felélesztésére, tesztelésére, az opció egyeztetésére és a vonalak elengedésére. LCP protokoll. (szinkron/aszinkron áramkörök, bájtalapú/bit alapú kódolás)
 3. Olyan módot a hálózati réteg-opciók megbeszélésére, amely független az alkalmazott hálózati réteg-protokolltól. Külön-külön NCP protokollok mindegyik támogatott hálózati réteghez.



PPP 2/2

36

- bájt alapú keretszerkezet, azaz a legkisebb adategység a bájt
- Alapértelmezésben nem biztosít megbízható átvitelt.
- **Mezők fontosabb tulajdonságai**
 - ▣ Vezérlő mező alapértéke a számozatlan keretet jelzi
 - ▣ Protokoll mezőben protokoll kód lehet az LCP, NCP, IP, IPX, AppleTalk vagy más protokollhoz.
 - 0-s bittel kezdődő kódok a hálózati protokollokhoz tartoznak (IP, XNS, stb.)
 - 1-es bittel kezdődő kódok a további egyeztetést igénylő protokollokhoz tartoznak (LCP, NCP, stb.)

1	1	1	1 vagy 2	változó	2 vagy 4	1
Jelző 01111110	Cím 1111111	Vezérlő 00000011	Protokoll	Adatmező	Ellenőrző összeg	Jelző 01111110

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						

Köszönöm a figyelmet!