

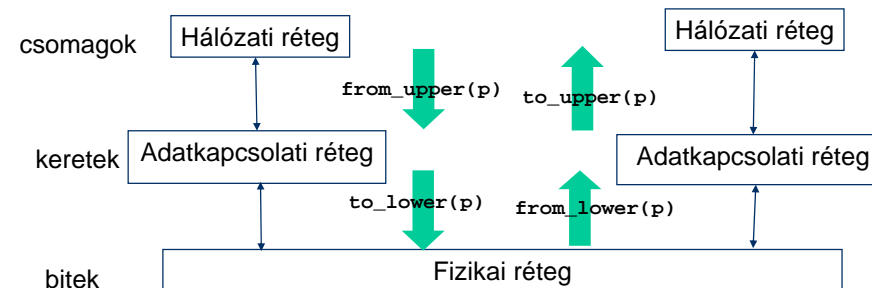
Számítógépes Hálózatok és Internet Eszközök

2007

9. Adatkapcsolati réteg – utólagos hibajavítás, csúszó ablakok

Utólagos hibajavítás

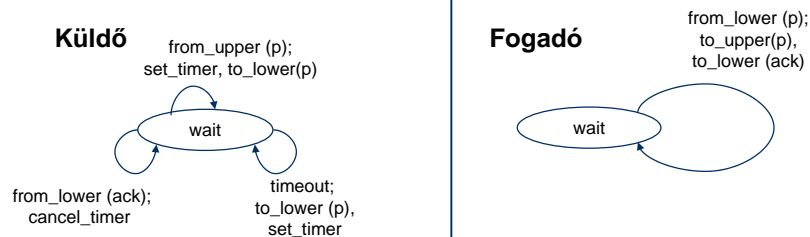
- A hiba felismerésekor a keretet újra kell küldeni
- Hogy néz ki a küldő és a fogadó összehangolt munkája?



to_lower, from_lower tartalmazzák a CRC-t
vagy (szükség esetén) utólagos hibajavítást

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:

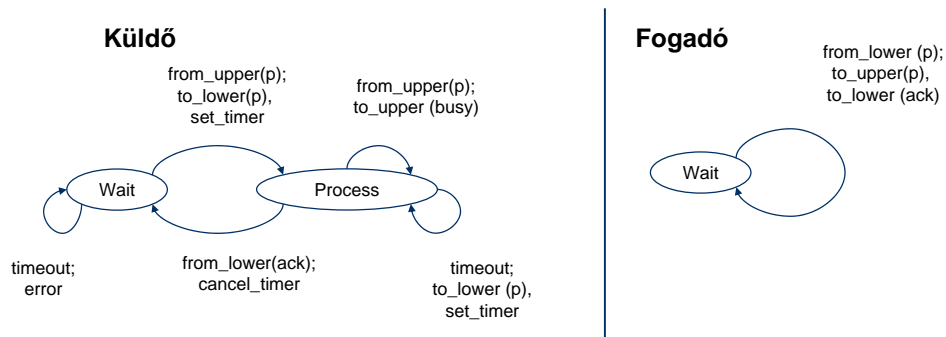


Elemzés

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

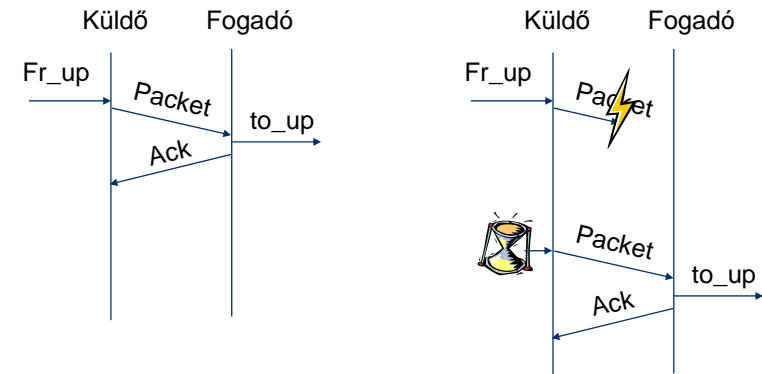
2. Kisérlet

- Az első probléma megoldása
 - Egy csomag a másik után



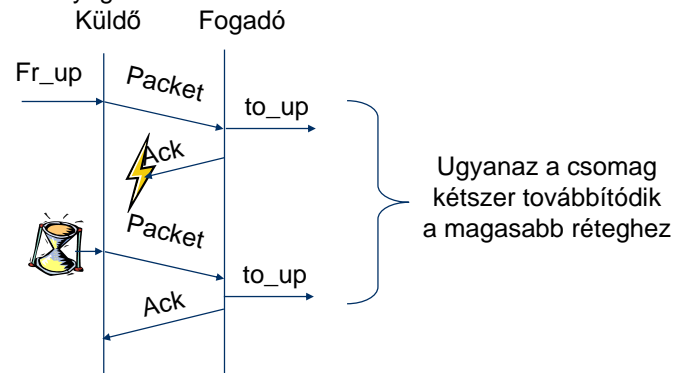
Elemzés

- A protokoll megvalósít egy elemi folyamfelügyeletet



Elemzés

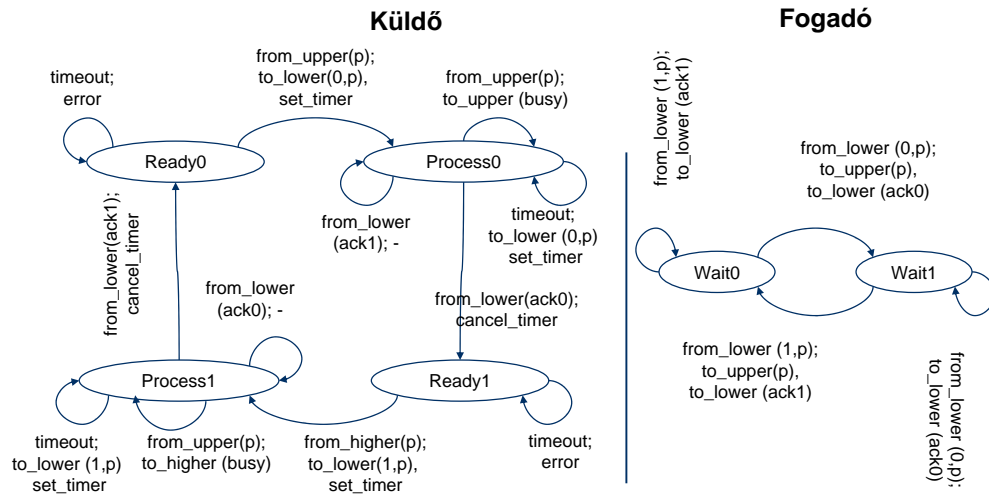
- 2. probléma: elveszik a nyugta



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

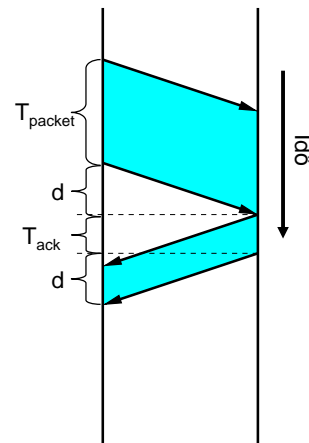


3. kísérlet: alternáló bit protokoll (Alternating Bit Protocol)

- A 3. kísérlet egy zajos csatorna fölötti megbízható protokoll korrek 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

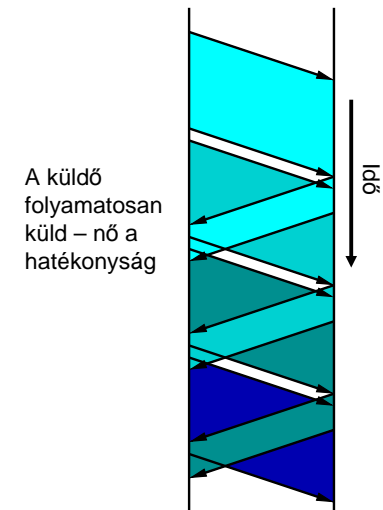
Alteráló bit protokoll -- hatékonyság

- 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

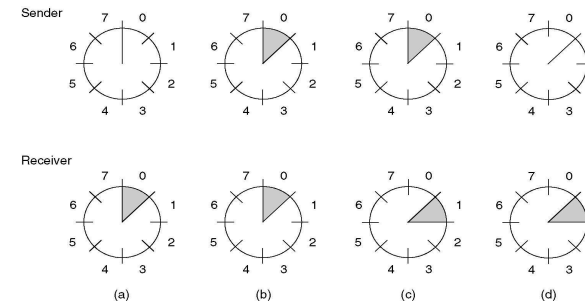


Csúszó ablak (sliding window)

- 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

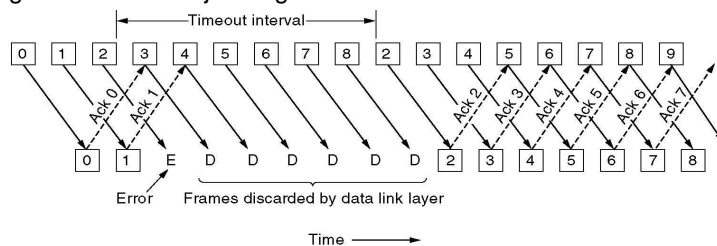
- “Sliding Window” példa $n=3$ és fix ablakméret = 1 esetén
- A küldő itt mutatja a még nem nyugtázott sorozatszámokat
 - Ha a még nem nyugtázott keretek (frame) száma ismert, akkor ez ekvivalens az előző főlán definiált a küldő-ablakkal



- 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

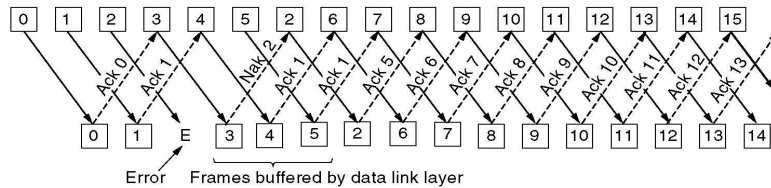
- Ha a fogadó-ablakméret = 1, akkor a fogadó nem tudja feldolgozni azokat a frame-eket, melyek egy elveszett (vagy hibás) frame-et követnek
 - Nem tudja azokat nyugtázni, mert csak egy nyugtát küld az utolsó helyesen fogadott csomagról
- A küldőnél lejár a várakozási idő a nyugtára: “Timeout”
 - Minden 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 közbenső 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**)