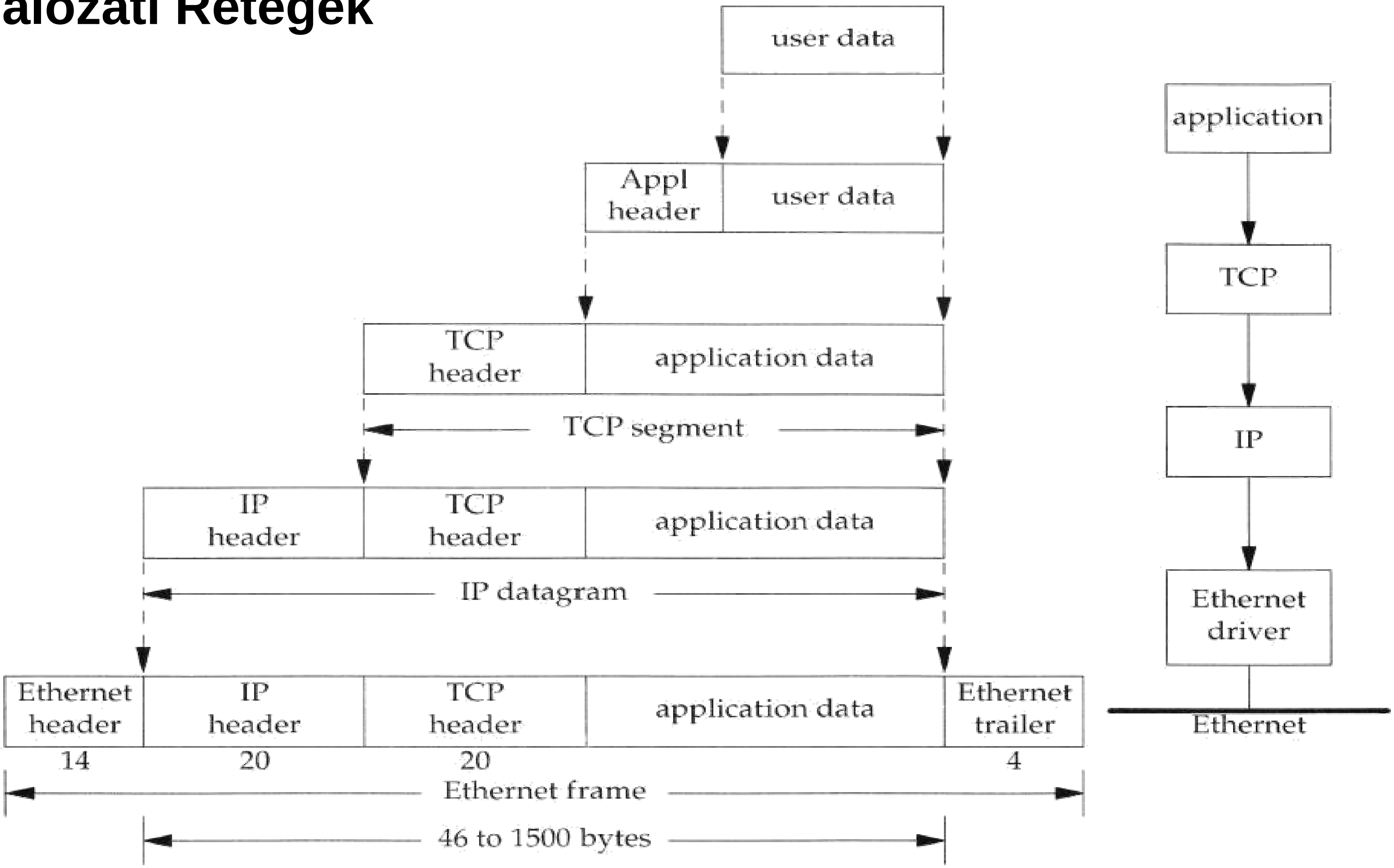
Hálózatok II
2006

4: TCP – Hatékonyság, Fairness

Hálózati Rétegek



TCP Működése

TCP (Transmission Control Protocol) egy kapcsolatorientált megbízható szolgálat bidirekcionális byte-folyamokhoz

TCP Kapcsolatorientált

- Két résztvevő. Egy egy résztvevő socket által azonosított:
  socket: IP-cím és port
- TCP-kapcsolat egyértelműen azonosított egy socketpár által
- Nincs broadcast sem multicast
- Kapcsolatfelépítés és lezárás szükséges
- Amíg egy kapcsolat nincs (rendesen) lezárva, addig akvív

TCP Működése

TCP egy kapcsolatorientált megbízható szolgálat bidirekcionális byte-folyamokhoz

TCP megbízható

- Minden adatcsomag megérkezését megerősíti *(acknowledgment)*
- A nem megerősített adatcsomagokat újraküldi
- “Checksum” a fejléchez és csomaghoz
- TCP számozza a csomagokat és sorbarendezi a fogadónál
- Törli a duplikált csomagokat

TCP Működése

TCP egy kapcsolatorientált megbízható szolgálat bidirekcionális byte-folyamokhoz

TCP egy szolgálat bidirekcionális byte-folyamokhoz

- Az adatok két egymással ellentétes irányú byte-sorozatként (=8 bit) kerülnek átvitelre
- A tartalom nem interpretálódik
- Az adatcsomagok időbeli viselkedése különbözhet: átvitel sebessége növekedhet, csökkenhet, más késés, más sorrendben is megérkezhetnek
- Megpróbálja az adatcsomagokat időben egymáshoz közel kiszállítani
- Megpróbálja az átviteli médiumot hatákonyan használni

  = kevés csomag

TCP-Fejléc (I)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- „Checksum“
  - Fejléchez és adatokhoz
- Fejléchossz (data offset)
  - A változzó hosszúságú opció-mező miatt

- Küldő-Port + Cél-Port-Nr.
  - Megenged több TCP-kapcsolatot IP-címenként
- Sorozatszám
  - Minden adatbyte meg van számozva modulo 2^{32}-1
  - = a szegmens első byte-jának a száma
- Megerősítés szám
  - Aktivivált az ACK-Flag által
  - Az első még nem feldolgozott adatbyte száma
  - = utolsó sorozatszám + utolsó adatmennyiség

TCP-Fejléc (II)

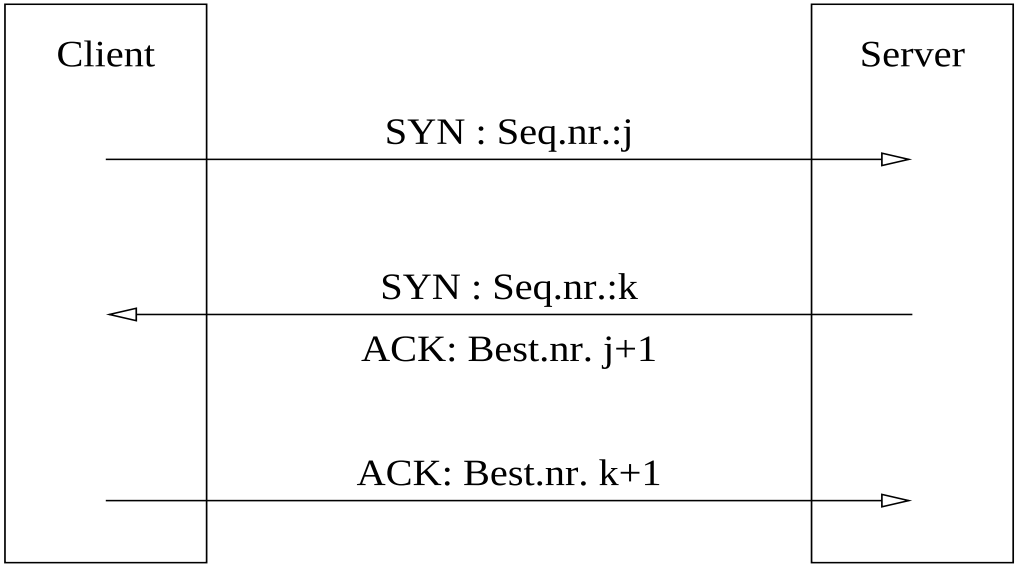
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Opció-mező pl. MSS (maximum segement size):
  - A fogadó megadja a kívánt csomagméretet
  - →Tekintet nélkül az IP MTU-ra (max. transmission unit)
  - → Fragmentálás lehetséges a az IP által

- FLAGS (függetlenül felhasználhatók)
  - `URG`: sürgős (urgent)
  - `ACK`: megerősítés (acknowledgment)
    - Aktiviálja a megerősítés számot
  - `PSH`: Push
    - Gyors adattovábbítás a felhasználói rétegnek
  - `RST:` Reset
    - A válasz hiba esetén: `connection reset by peer`
  - `SYN:`Synchronize
    - Kapcsolatfelépítés és az iniciális sorozatszám megadása
  - `FIN:` Finished
    - (Egy) adatfolyam befejezése

Kapcsolatfelépítés

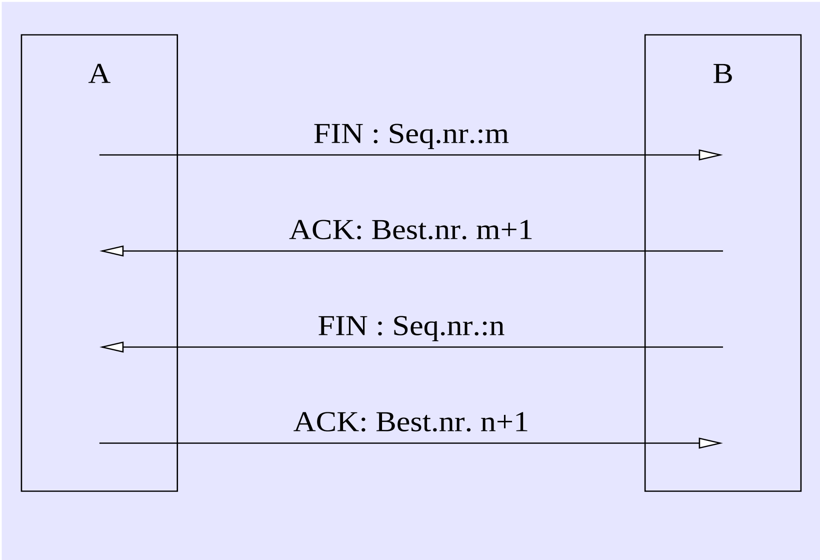
- Rendszerint Client-Server-kapcsolat
  - Ekkor felépítés 3 TCP-csomaggal (=3 szegmens)
  - Az első SYN-szegmensben az MSS (maximum segment size) is átvitelre kerül



Kapcsolat lezárása

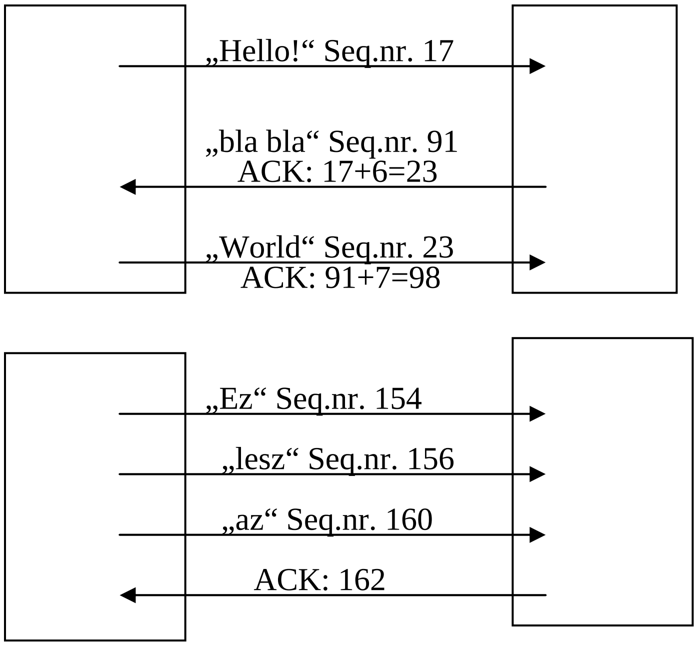
- Félig lezárás (half-close)
  - A küldő jelzi a kapcsolat befejezését egy FIN-szegmensben és vár annak megerősítésére
  - Az ellenkező irányban továbbra is lehet küldeni

- 2 Félig lezárás lezárja a TCP-kapcsolatot



Megerősítés (acknowledgement)

- „piggybacking“
  - A megerősítések az ellenkező irány adatcsomagjain „lovagolnak“
- Egy megerősítés több adatszegmenst is megerősíthet
  - Ha nincs küldeni való adat, késelteti az ACK-kat



Nagle Algoritmusa

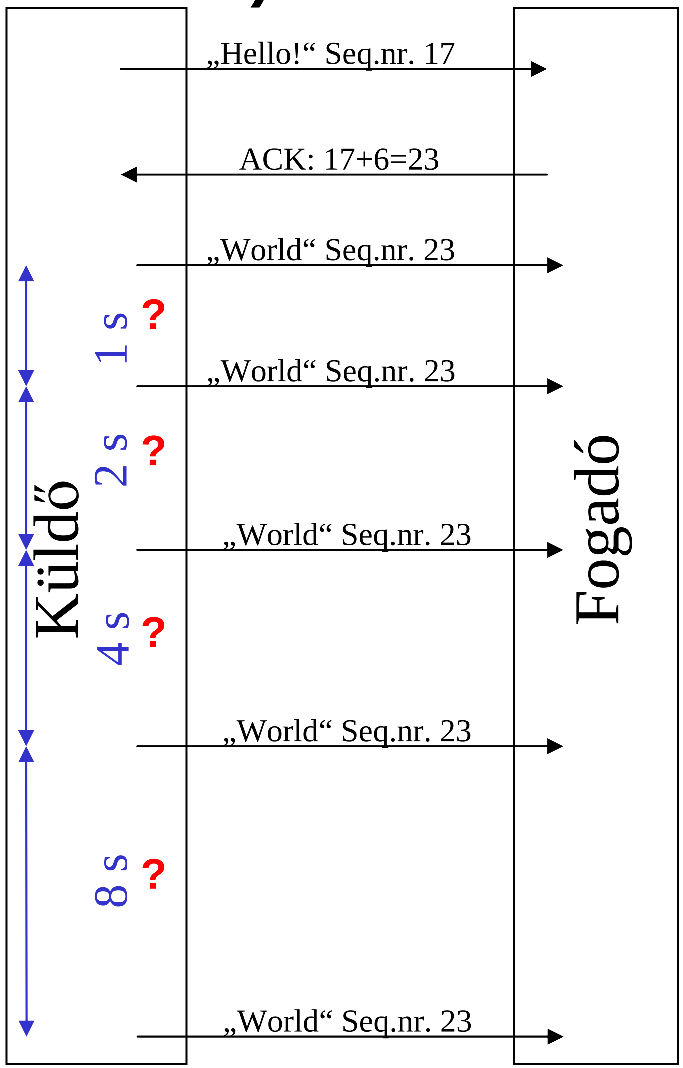
Kis csomagok nem kerülnek addig küldésre, amig megerősítések hiányoznak

- Önmagát ütemező: Gyors kapcsolat = sok kis csomag
- Egy csomag kicsi, ha az adathossz < MSS

Exponenciális visszavétel (exponential backoff)

- Retransmission Timout (RTO) szabályozza az időközt a küldés és egy duplikátum újraküldése között, ha egy megerősítés kimarad
- Mikor nem kerül megerősítésre egy TCP-csomag?
  - Ha a megerősítés sokkal több időt vesz igénybe, mint az átlagos “round trip time (RTT)”
    - 1. Probléma: RTT mérése
    - 2. Probléma: Csak a megerősítés jön túl későn
  - Küldő
    - Vár a Retransmission Timout-nek megfelő ideig
    - Ha nem érkezett megerősítés újraküldi a csomagot és növeli

      RTO ← 2 RTO   (RTO = 64 másodpercig)
- RTO újraszámolása, ha a csomagok megerősítésre kerülnek



A Round Trip Time (RTT) Becslése

- A TCP-csomag nem megerősítettnek számít, ha a megerősítés “lényegesen” tovább tart, mint az RTO
- RTT nem számítható on-line (csak visszatekintve)
- RTT erősen ingadozik
- Ezért: Retransmission Timeout Value nagyvonalú becsléssel:
  - RFC 793: (M := utoljára mért RTT)

    $R \leftarrow \alpha R + (1-\alpha) M,$ ahol $\alpha = 0{,}9$

    $RTO \leftarrow \beta R,$ ahol $\beta = 2$
  - Jacobson 88: a becslés nem elég robosztus, ezért

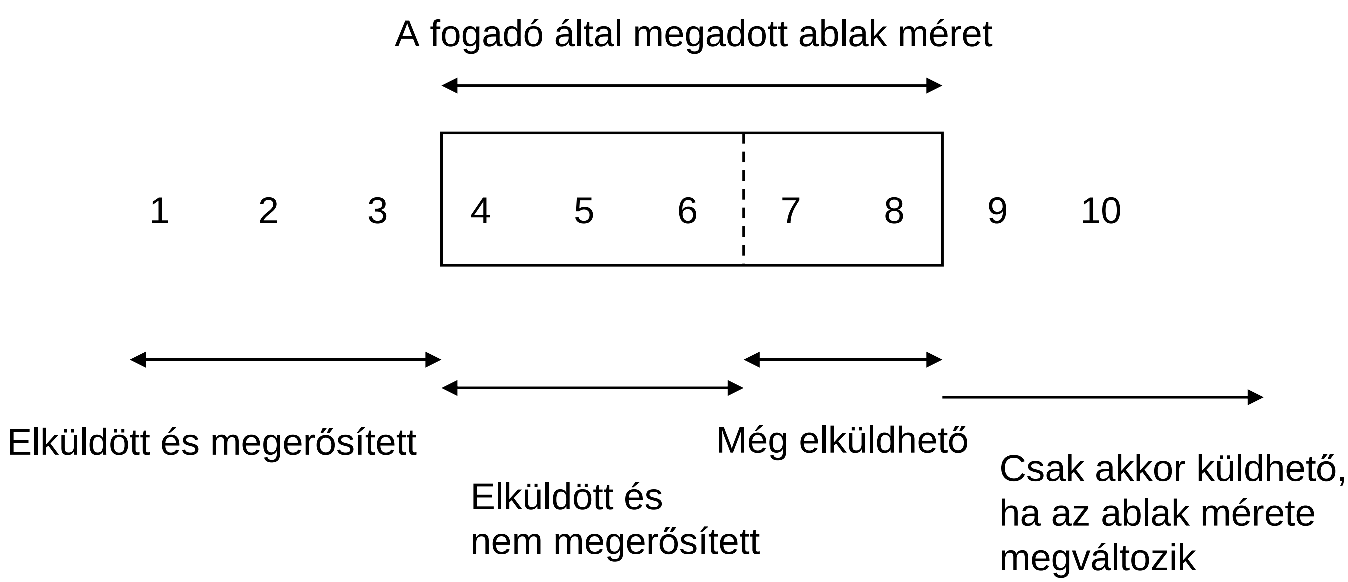
    $A \leftarrow A + g (M - A) ,$ ahol $g = 1/8$

    $D \leftarrow D + h (|M - A| - D) ,$ ahol $h = 1/4$

    $RTO \leftarrow A + 4D$
- Többszörösen elküldött csomagoknál nem aktualizálunk

Csúszó Ablakok (sliding windows)

- Adatátráta szabályozás ablak segítségével
  - A fogadó meghatározza az ablak méretet (*wnd*) az ACK-szegmensek TCP-fejlécében
  - Ha a fogadó fogadási pufferja tele van, *wnd=0* -t küld
  - Máskülönben a fogadó *wnd>0* -t küld
- A küldőnek be kell tartani:
  - Az elküldött nem megerősített adatok száma ≤ ablak mérete



Lassú Start (slow start)

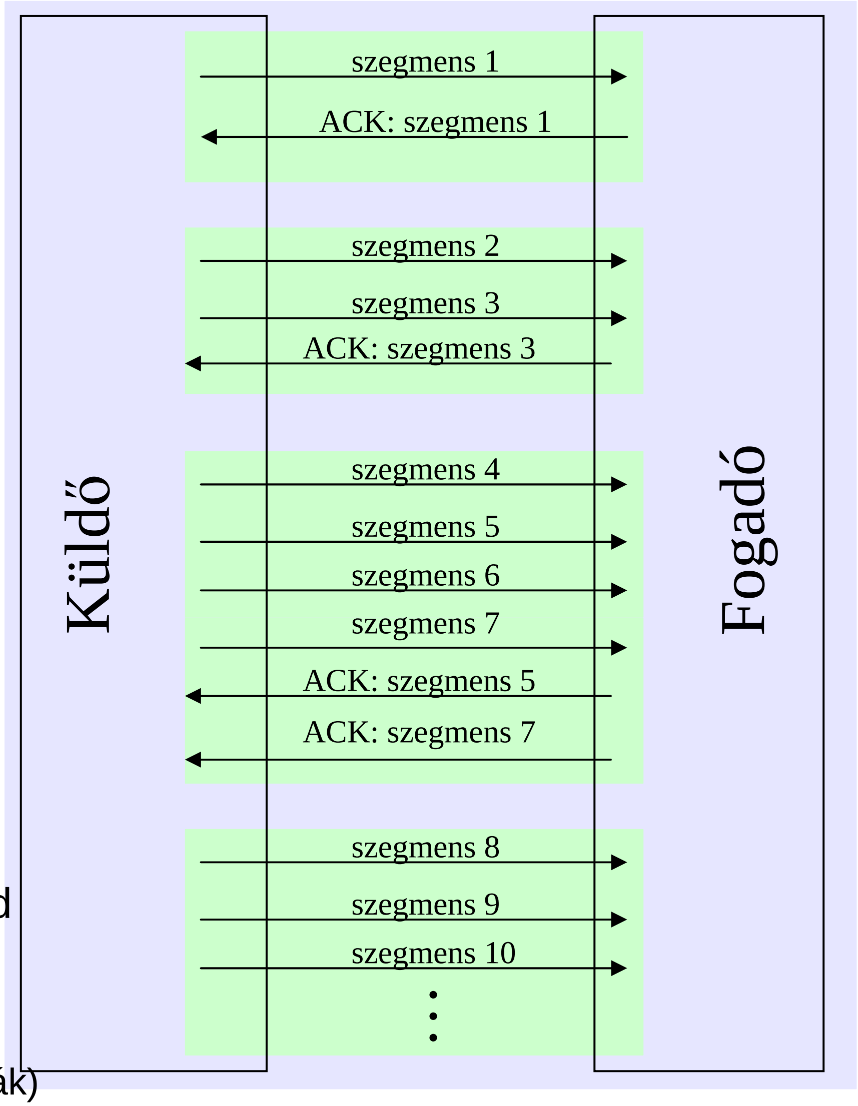
- A küldőnek nem szabad a fogadó által felajánlott ablakméretet azonnal kihasználni
- Második ablak: Congestion-ablak (cwnd / Congestion window)
  - A küldő választja (FSK)
  - Az ablak amiben küld: min {*wnd,cwnd*}
  - Kezdetben:

    $$\text{cwnd} \leftarrow \text{MSS}$$

  - Minden csomagnál a megkapott megerősítés után nő

    $$\text{cwnd} \leftarrow \text{cwnd} + \text{MSS}$$

    (azaz megduplázódik minden RTT után)
  - Addig, amíg egyszer egy megerősítés kimarad



Slow start = exponenciális növekedés
(hisztórikus elnevezés: korábban még aggresszívebb sémák)

Dugó elkerülés (congestion avoidance)

- Jacobson 88:
  - Parameter: *cwnd* és Slow-Start-küszöb (*ssthresh* = slow start threshold)

1. Kapcsolatfelépítés:

   $cwnd \leftarrow \text{MSS} \qquad ssthresh \leftarrow 65535$

2. Csomagvesztésnél, azaz megerősítás ideje > RTO: multiplicatively decreasing

   $cwnd \leftarrow \text{MSS} \qquad ssthresh \leftarrow \max\left\{2\,\text{MSS}, \frac{\min\{cwnd, wnd\}}{2}\right\}$

3. Megerősítés jön a szegmenshez és *cwnd* ≤ *ssthresh*: slow start

   $cwnd \leftarrow cwnd + \text{MSS}$

4. Megerősítés jön a szegmenshez és *cwnd* > *ssthresh*: additively increasing

   $cwnd \leftarrow cwnd + \frac{\text{MSS}}{cwnd}$

Fast Retransmit és Fast Recovery

- Jacobson 90: Ha csak egy csomag veszik el, akkor
  - A csomag és az ablak fennmaradó részének újraküldése
  - Egyidejűleg slow start

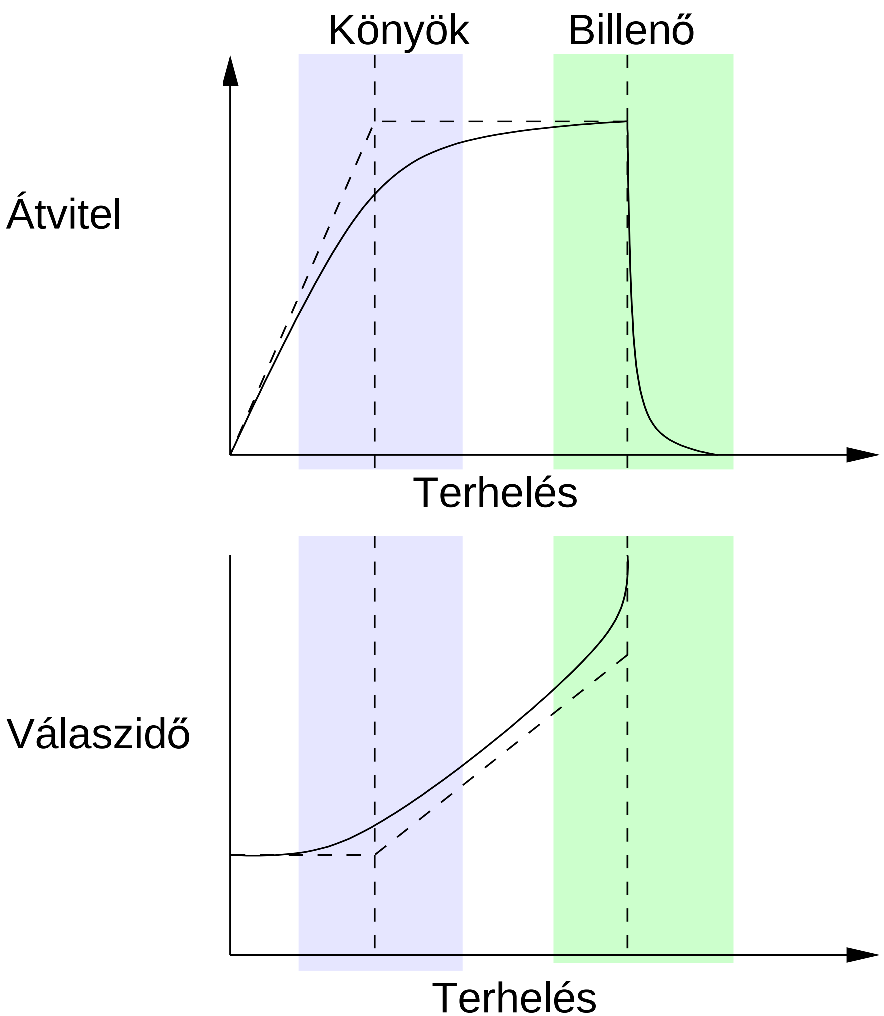
- Fast Retransmit: Ugyanazon csomag három megerősítése után (triple duplicate ACK), a csomagot újraküldjük
- Fast Recovery*:* Növeljük a csomagküldési rátát minden új megerősítésnél

1. Fast retransmit: Egy triple duplicate ACK után egy P csomaghoz, küldjük újra P-t
   $ssthresh \leftarrow \min(wnd, cwnd)/2$ $\quad$ $cwnd \leftarrow ssthresh + 3$ MSS

2. Fast recovery: P újabb megerősítésénél: $cwnd \leftarrow cwnd +$ MSS

3. Congestion avoidance: Ha megérkezik P+x megerősítése: $cwnd \leftarrow ssthresh$

Additive Increase Multiplicative Decrease (AIMD): Fairness és Hatékonyság



A hálózati terhelés az átvitellel és a válasziővel kölcsönösen hat egymásra.

- Az átvitel maximális, ha a terhelés a hálózat kapacitását majdnem eléri.
- Ha a terhelés tovább nő, túlcsordulnak a pufferek, csomagok vesznek el, újra kell küldeni, drasztikusan nő a válaszidő. Ezt az "adatdugót" congestion-nak nevezzük.
- Ezért a maximális terhelés helyett, ajánlatos a hálózat terhelését a könyök közelében beállítani. Itt a válaszidő csak lassan emelkedik, míg az adatátvitel már a maximum közelében van.
- Egy jó dugóelkerülési (*congestion avoidance*) stratégia a hálózat terkelését a könyök közlében tartja: hatékonyság. Emellett fontos, hogy minden résztvevőt egyforma rátával szolgáljunk ki: fairness.

AIMD Fairness és Hatékonyság – A Modell

- n résztvevő, forduló-modell
- résztvevő i adatrátája a t-eik fordulóban $x_i(t)$
- Kezedeti adatráták: $x_1(0), \ldots, x_n(0)$
- A visszacsatolás (feedback) forduló t után: y(t) = 0, ha $\sum_{i=1}^{n} x_i(t) \leq K$

  y(t) = 1, ha $\sum_{i-1}^{n} x_i(t) > K$
- Minden résztvevő aktualizálja az adatrátáját a t+1-edik fordulóban:

  $x_i(t+1) = f(x_i(t),y(t))$

  - Increase-stratégia $f_0(x) = f(x,0)$
  - Decrease-stratégia $f_1(x) = f(x,1)$
- Tekintsük a következő lineáris függvényeket :

$$f_0(x) = a_I + b_I x, \qquad f_1(x) = a_D + b_D x$$

AIMD Fairness és Hatékonyság– A Modell

- A következő lineáris függvényeket vizsgáljuk:

$$f_0(x) = a_I + b_I x, \qquad f_1(x) = a_D + b_D x$$

- Érdekes speciális esetek:
  - MIMD: Multiplicative Increase/Multiplicative Decrease

$$f_0(x) = b_I x, \quad f_1(x) = b_D x, \quad \text{ahol} \quad b_I > 1,\ b_D < 1.$$

  - AIAD: Additive Increase/Additive Decrease

$$f_0(x) = a_I + x, \quad f_1(x) = a_D + x, \quad \text{ahol} \quad a_I > 0,\ a_D < 0.$$

  - AIMD: Additive Increase/Multiplicative Decrease

$$f_0(x) = a_I + x, \quad f_1(x) = b_D x, \quad \text{ahol} \quad a_I > 0,\ b_D < 1.$$

AIMD Fairness és Hatékonyság

- Hatékonyság
  - Terhelés: $X(t) := \sum_{i=1}^{n} x_i(t)$
  - Mérték: $|X(t) - K|$

- Fairness: $x=(x_1, \ldots, x_n)$ esetén:

$$F(x) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} (x_i)^2} .$$

  - $1/n \le F(x) \le 1$
  - $F(x) = 1 \leftrightarrow$ absolut Fairness
  - skálázástól Független
  - Folytonos, differenciálható
  - Ha n közül k fair, a többi 0, akkor $F(x) = k/n$

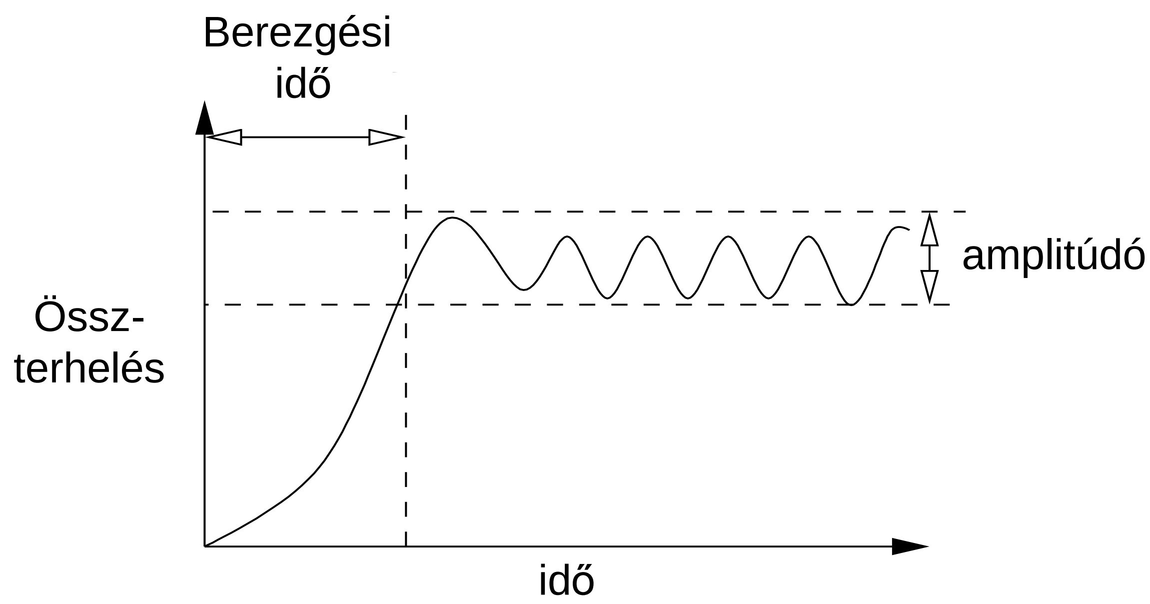
Konvergencia

- Konvergencia nem lehetséges
- Legjobb esetben oszcilláció az optimális érték körül
  - Az oszcilláció amplitúdója A

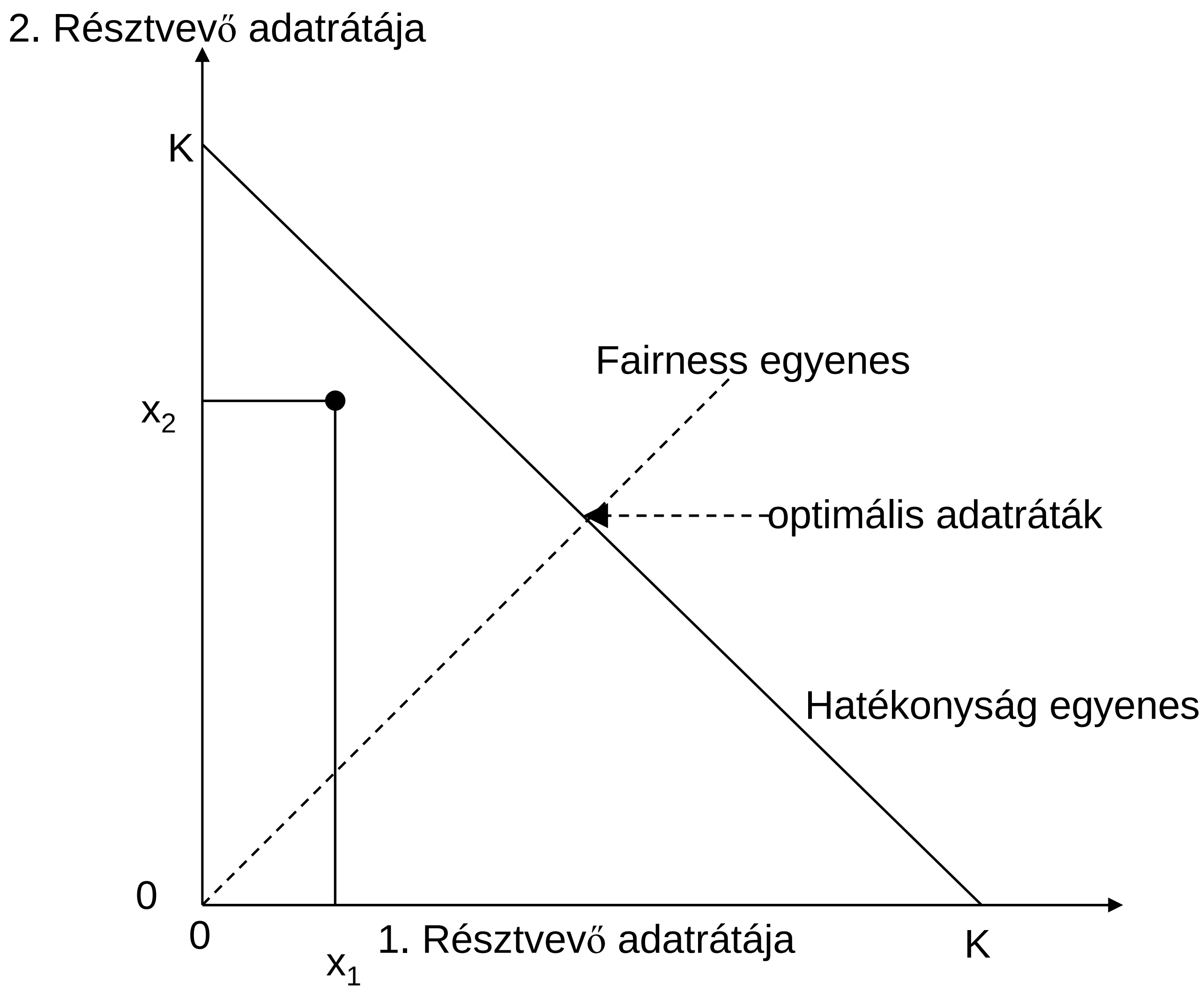
$$A = \inf_{t_0 \geq 0} \sup_{t \geq t_0} |X(t) - K| \ .$$

  - Berezgési idő T

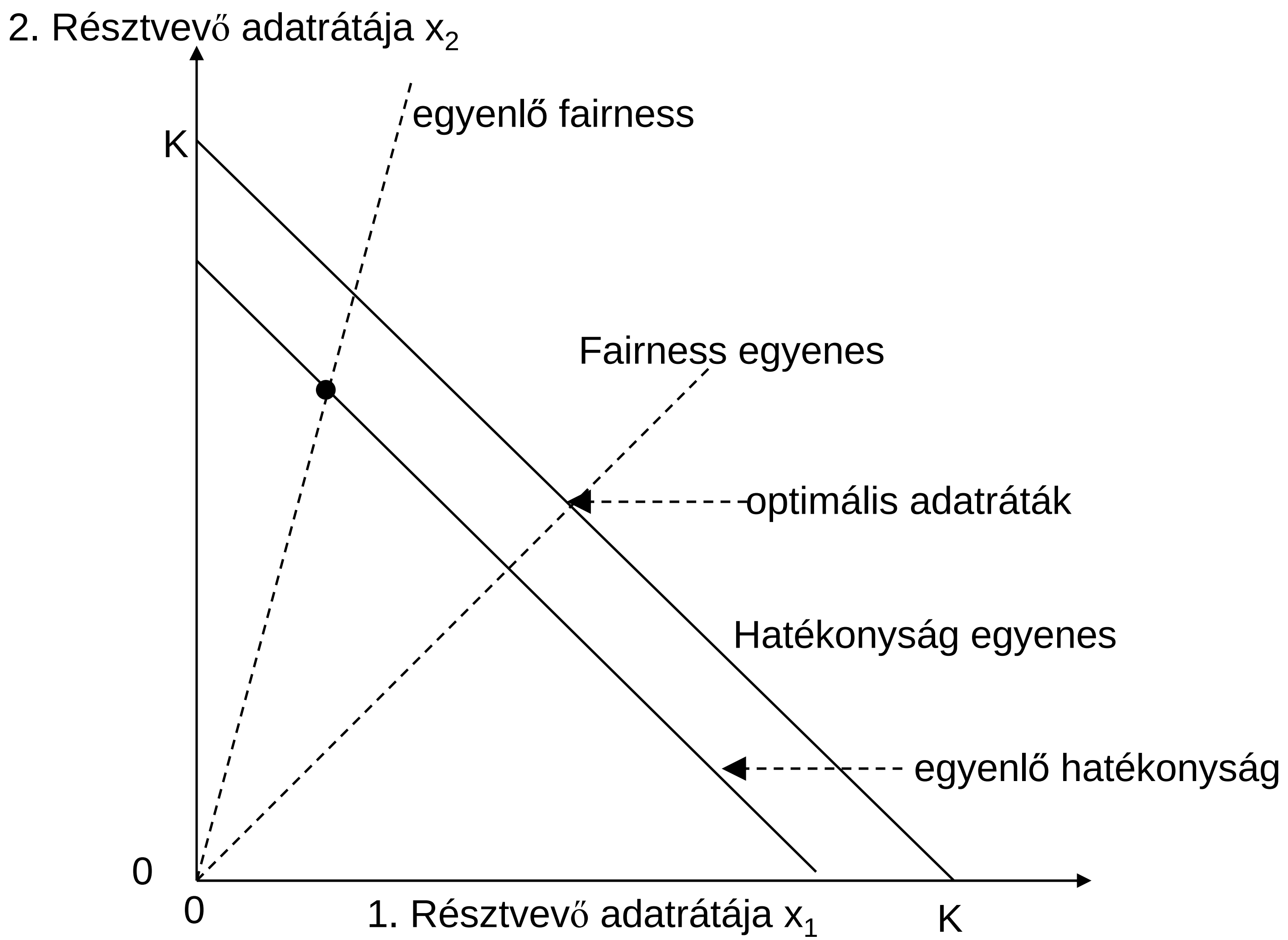
$$T = \min\{t_0 \mid \forall t \geq t_0 \ : \ |X(t) - K| \leq A\} \ .$$



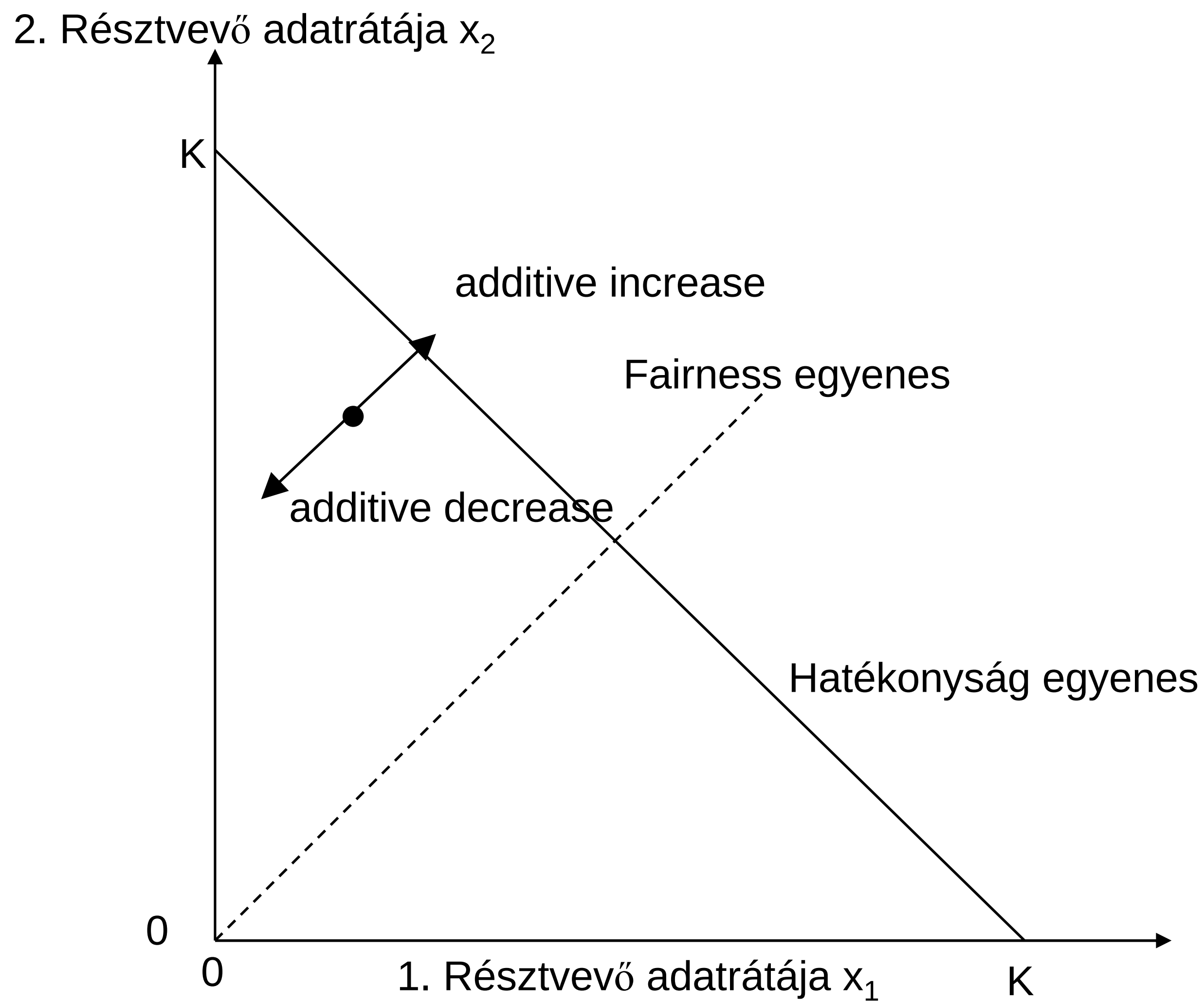
Vektor Ábrázolás (I)



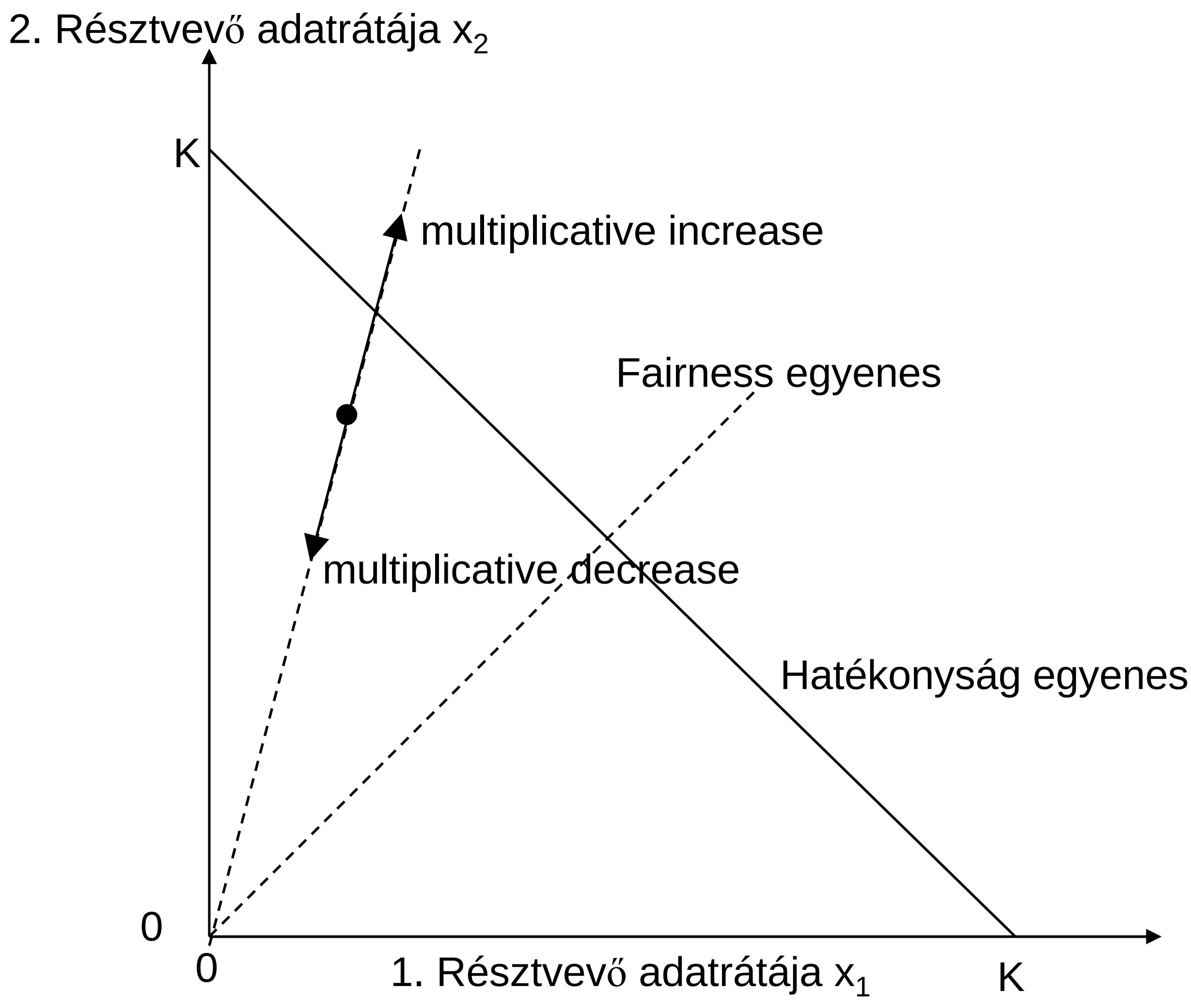
Vektor Ábrázolás (I)



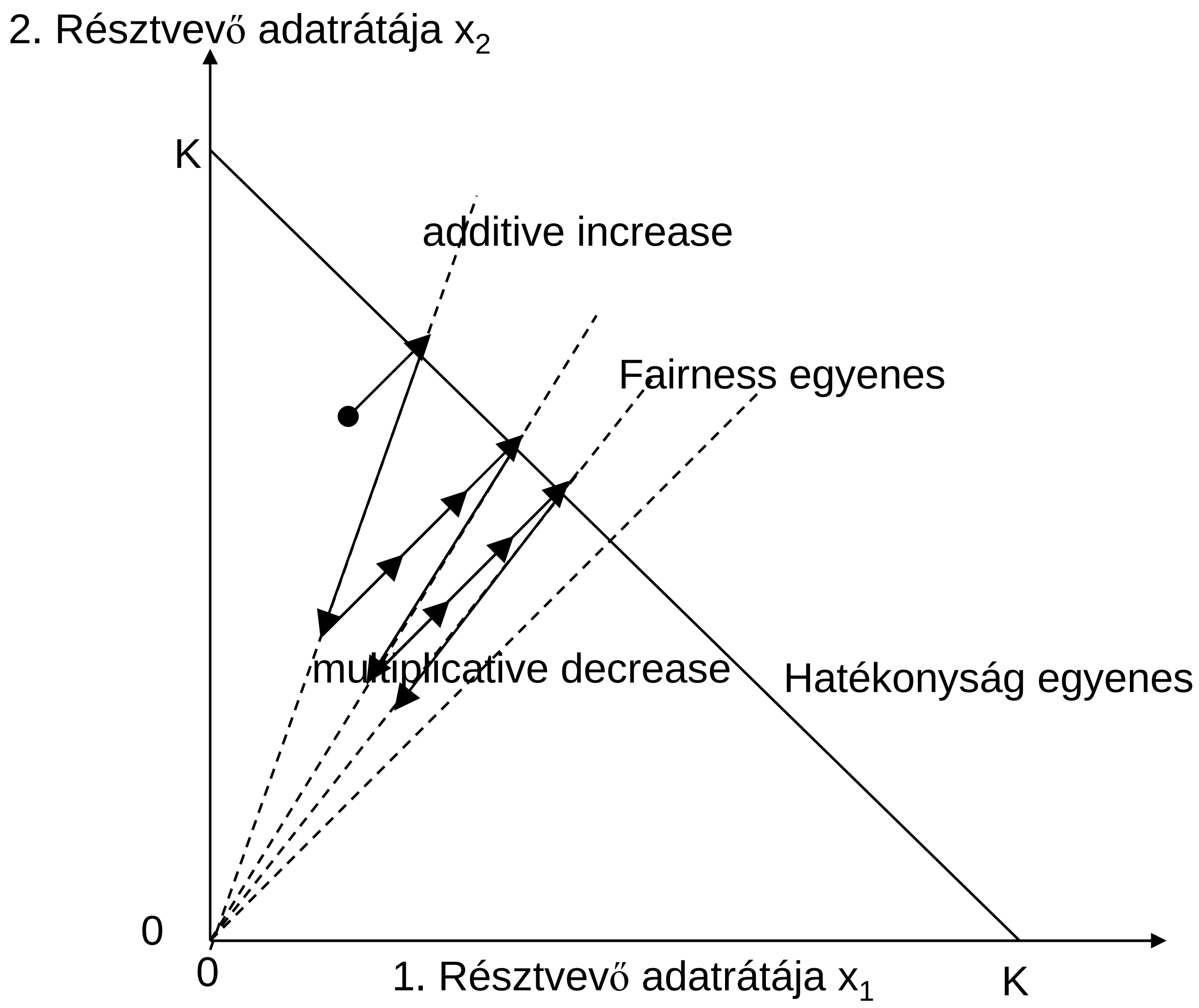
Vektor Ábrázolás (I)



Vektor Ábrázolás (I)



Vektor Ábrázolás (I)



Hatékony Lineáris Függvények

- $X(t) > K$

$$
\begin{aligned}
X(t+1) < X(t) &\iff \sum_{i=1}^{n} x_i(t+1) < \sum_{i=1}^{n} x_i(t) \\
&\iff \sum_{i=1}^{n} a_D + b_D x_i(t) < \sum_{i=1}^{n} x_i(t) \\
&\iff n a_D + (b_D - 1) X(t) < 0 \\
&\iff b_D < 1 - \frac{n a_D}{X(t)} .
\end{aligned}
$$

  - $a_D \le 0 \rightarrow b_D \le 1$
  - $a_D > 0 \rightarrow b_D < 0$

- $X(t) < K$

$$
\begin{aligned}
X(t+1) > X(t) &\iff \sum_{i=1}^{n} x_i(t+1) > \sum_{i=1}^{n} x_i(t) \\
&\iff \sum_{i=1}^{n} a_I + b_I x_i(t) > \sum_{i=1}^{n} x_i(t) \\
&\iff n a_I + (b_I - 1) X(t) > 0 \\
&\iff b_I > 1 - \frac{n a_I}{X(t)} .
\end{aligned}
$$

  - $a_I \ge 0 \rightarrow b_I \ge 1$
  - $a_I < 0 \rightarrow$ nem lehet

Fairness (I)

- Az x(t) = (x_1(0), …, x_n(0)) fairnesse konvergál 1-hez, azaz

$$\lim_{t\to\infty} F(x(t)) = 1 \ .$$

ahol

$$F(x) \ = \ \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n}(x_i)^2} \ .$$

- Teljesül c=a/b esetén:

$$F(x(t+1)) = F(x(t)) + (1 - F(x(t))) \left(1 - \frac{\sum_{i=1}^{n}(x_i(t))^2}{\sum_{i=1}^{n}(c + x_i(t))^2}\right)$$

- Bizonyítás?

Bizonyítás! (1)

- Teljesül:

$$F(x(t+1)) = \frac{\left(\sum_{i-1}^{n} x_i(t+1)\right)^2}{n\sum_{i=1}^{n}(x_i(t+1))^2} = \frac{\left(\sum_{i-1}^{n} a + bx_i(t)\right)^2}{n\sum_{i=1}^{n}(a+bx_i(t))^2}$$

$$= \frac{\left(\sum_{i=1}^{n} \frac{a}{b} + x_i(t)\right)^2}{n\sum_{i=1}^{n}\left(\frac{a}{b}+x_i(t)\right)^2} = \frac{\left(\sum_{i=1}^{n} c + x_i(t)\right)^2}{n\sum_{i=1}^{n}(c+x_i(t))^2}$$

- Behelyettesítve

$$X = \sum_{i=1}^{n} x_i(t)\,, \qquad Q = \sum_{i=1}^{n} (x_i(t))^2\,,$$

$$L = \sum_{i=1}^{n} (x_i(t)+c)\,, \quad R = \sum_{i=1}^{n} (x_i(t)+c)^2\,.$$

- Ekkor: $F(x(t)) = \frac{X^2}{nQ}$ és $F(x(t+1)) = \frac{L^2}{nR}$

Bizonyítás! (2)

- Meg kell mutatni: $F(x(t+1)) \;=\; F(x(t)) + (1 - F(x(t)))\left(1 - \frac{Q}{R}\right)$

$$
\begin{aligned}
\Longleftrightarrow \qquad \frac{L^2}{nR} &= \frac{X^2}{nQ} + \left(1 - \frac{X^2}{nQ}\right)\left(1 - \frac{Q}{R}\right) \\
\Longleftrightarrow \qquad QL^2 &= RX^2 + \left(nQ - X^2\right)(R - Q) \\
\Longleftrightarrow \qquad QL^2 &= RX^2 + nQR - nQ^2 - RX^2 + QX^2 \\
\Longleftrightarrow \qquad L^2 &= nR - nQ + X^2 \\
\Longleftrightarrow \qquad L^2 - X^2 &= n(R - Q)\,.
\end{aligned}
$$

- Miért érvényes ez az egyenlőség?

Bizonyítás! (3)

- Miért érvényes ez az egyenlőség?

$$L^2 - X^2 = n(R - Q)$$

ahol

$$X = \sum_{i=1}^{n} x_i(t)\ , \qquad Q = \sum_{i=1}^{n} (x_i(t))^2\ ,$$
$$L = \sum_{i=1}^{n} (x_i(t) + c)\ , \quad R = \sum_{i=1}^{n} (x_i(t) + c)^2\ .$$

$$n(R-Q) = n\sum_{i=1}^{n}(x_i(t)+c))^2 - x_i(t)^2 = n\sum_{i=1}^{n}(2cx_i(t)+c^2) = 2cnX + c^2n^2$$

$$L^2 - X^2 = \sum_{i,j\in[1..n]} (x_i(t)+c)(x_j(t)+c) - x_i(t)x_j(t)$$
$$= \sum_{i,j\in[1..n]} cx_i(t) + cx_j(t) + c^2 = 2cnX + c^2n^2$$

Fairness (II)

- Teljesül, hogy:
$$F(x(t+1)) = F(x(t)) + (1 - F(x(t)))\left(1 - \frac{\sum_{i=1}^{n}(x_i(t))^2}{\sum_{i=1}^{n}(c + x_i(t))^2}\right)$$
- Azaz a fairness növekszik, ha c=a/b növekszik.
  - c=0, akkor F(x(t+1))= F(x(t))
  - c>0, akkor F(x(t+1))> F(x(t)), ha F(x(t)) ≠ 1
  - c<0 ist F(x(t+1))< F(x(t))
- Így $a_I/b_I \geq 0$ és $a_D/b_D \geq 0$
  - tehát, $a_I, b_I, a_D, b_D \geq 0$
- A hatákonyságból:
  - $a_D \leq 0 \rightarrow b_D \leq 1$, azt jelenti tehát, hogy $a_D = 0 \rightarrow b_D \leq 1$,
  - $a_D > 0 \rightarrow b_D < 0$ kiesik.
  - $a_I \geq 0 \rightarrow b_I \geq 1$,
    - Teljesülni kell, hogy $a_I > 0$, máskülönben nem növekszik a fairness (lásd MIMD)
- Ez AIMD-hez vezet.

Fairness (III)

- Tétel 1: Ha f_0 és f_1 lineáris függvények, akkor a fairnesshez és a hatékonysághoz $x \geq 0$ esetén teljesülni kell, hogy:

$$
\begin{aligned}
f_0(x) &\geq x + a_I, && \text{ahol } a_I > 0 \text{ és} \\
f_1(x) &= b_D x, && \text{ahol } 0 < b_D < 1.
\end{aligned}
$$

TCP Adatráta

- AIMD probáló stratégia ℓ csomagvesztés-rátát okoz
- ℓ = elvesztett szegmensek / küldött szegmensek
- Közepes adatráta B = byte/sec. és a hibaráta ℓ kölcsönösen kihatnak egymásra:
  - Az adatráta növelése növeli a vesztésrátát
  - Magasabb vesztésráta csökenti az adatrátát
- Kisérletek mutatják, hogyha a szegmenshossz MSS és a „round trip time" RTT:

$$B = 1,3 \ \frac{\mathrm{MSS}}{\mathrm{RTT}\sqrt{\ell}}$$

TCP-Adatráta – A Statikus Eset

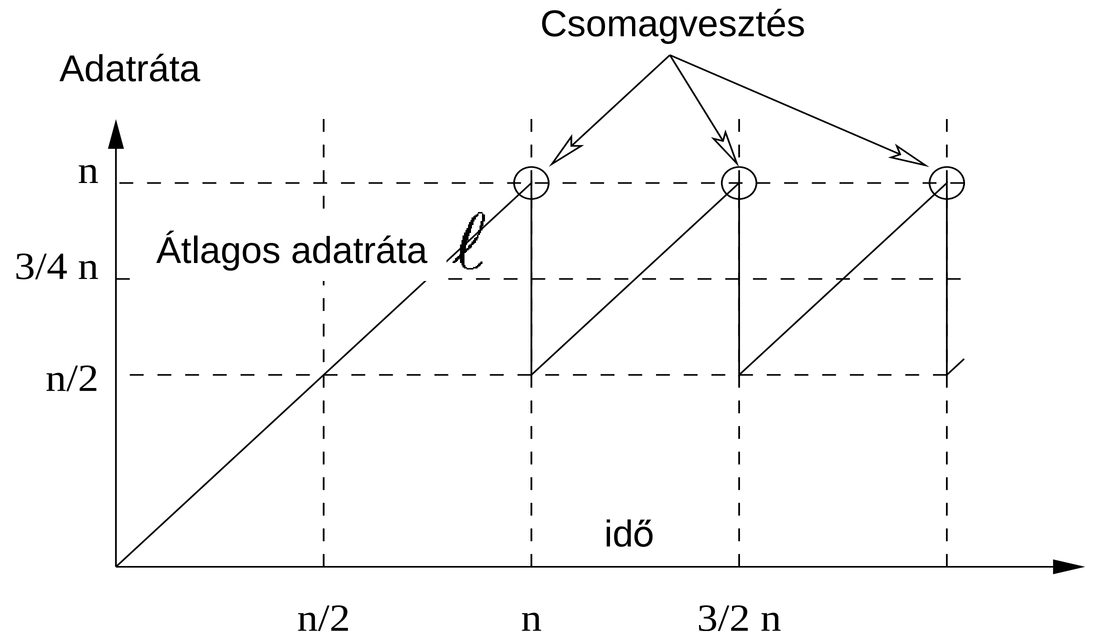
- n: rendelkezésre álló sávszélesség
- Átlagos adatráta:

  B = 3/4 n
- n/2 forduló után a veszlés: 1
- Így

$$\ell = \frac{1}{\frac{3}{4}n \cdot \frac{1}{2}n} = \frac{8}{3}\frac{1}{n^2}$$

- tehát $n = \sqrt{\frac{8}{3\ell}}$

$$B(\ell) = \frac{3}{4}\sqrt{\frac{8}{3\ell}}\frac{\mathbf{MSS}}{\mathbf{RTT}} = 1,22\ldots \cdot \frac{\mathbf{MSS}}{\mathbf{RTT}\sqrt{\ell}}$$



TCP-Adatráta – Egy Sztochasztikus Modell

- Egy csomag p valószínűséggel veszik el
- A hiba nélkül átvitt csomagok száma X exponenciális eloszlású:

$$\mathrm{P}[X=i] \;=\; p(1-p)^i$$

- A hiba nélkül küldött csomagok mennyiségének várható értéke : O(1/p)
  - A fordulók száma az adatráta felezéséig: $O(1/p^{1/2})$
  - Várható adatráta: $O(1/p^{1/2})$
  - De: Mi a konstans faktor?
- Kisérletileg: (rokon modellre bizonyított):

$$B(p) \quad = \quad 1,3\ldots \cdot \frac{\mathrm{MSS}}{\mathrm{RTT}\sqrt{p}}$$

Irodalom

- Van Jacobson: Congestion Avoidance and Control. In: *ACM SIGCOMM*, 314-329, 1988.
- Dah-Ming Chiu, Raj Jain: Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, Vol.: 17, 1-14, 1989.