Csomagküldés zajos csatornán

Biteket küldünk 1-esével pl. rádión vagy lézer-impulzusokkal,
pl. egy műholdra, vagy csak a szomszéd szobába.

Időegységenként 1 bitet tudunk küldeni — legyen most az időegység
$1\,\mu s$, vagyis a küldött mennyiség $1\,\frac{bit}{\mu s} = 10^6\,\frac{bit}{s}$

t_0

A bitek néha hibásan érkeznek meg, ilyenkor újra el kell küldeni őket.

Az adatot csomagokban küldjük, pl.

szinkronizáló mező	cím mező	sorszám mező	adatcsomag	hibajelző mező

$$\underbrace{\qquad\qquad\qquad}_{\text{csomag} = \text{keret} = \text{frame}}$$

A hibajelző mező tartalmát a csomag többi része alapján számoljuk.

A csomag érkezése után a vevő ellenőrzi a hibajelző mezőt (pl. újra kiszámolja). Ha nem stimmel, akkor hiba történt.

→ Ha stimmel, a vevő nyugtát küld a feladónak a csomagról („pozitív nyugta”)

→ Ha nem stimmel, akkor küldhet „negatív nyugtát” vagy semmit (ez a modellünk szempontjából mindegy).

A nyugtára adott ideig várunk.

Ha nem jön pozitív nyugta (vagyis vagy negatív jön, vagy semmi), akkor az egészt csomagot újraküldjük.

Egyszerűsítő ~~feltevések~~ $\boxed{\text{KÖZELÍTÉSEK:}}$

(1) Minden elküldött bit, a többitől függetlenül, azonos p_b val.séggel hibásodik meg. p_b kicsi.

[A valóságban ez nem igaz, pl. légköri zavar esetén egy csomó bit egymás után hibás lehet.]

(2) Minden hibára fény derül.

[A valóságban ez sem igaz: A hibajelző mező (tipikusan CRC = cyclic redundancy check) valahány hibáig megbízható, de ha túl sok a hiba (vagyis nagyon kicsi val.séggel), ~~akkor~~ véletlenül is stimmelhet.]

(3) A nyugtaküldés hibamentes.

[Vagyis nyugta nem vész el és nem alakul át, ~~Ez leh.~~ csak ~~nagyon kis~~ elhanyagolható val.séggel. Ez lehet akár életszerű feltevés is, mert a nyugta rövid, így nem nagy luxus redundánsan (pl. hibajavító kóddal) vagy többször is elküldeni.]

(4) Hibajelző kód van, de hibajavító kód nincs: ha egyetlen bit is hibás, muszáj az egészt csomagot újraküldeni.

[Ez is életszerű: a hibajavító kódok sok helyet foglalnak, és ha a hiba ritka, jobban megéri megismételni néhány csomagot.]

(5) Egy csomagot akárhányszor újraküldhetünk, amíg csak sikeresen át nem megy.

[A valóságban persze van egy limit, ami után a küldés hibaüzenettel leáll.]

(6) A csomag N db adat-bitből áll (hasznos adat)
+ M db kísérő-bitből (szinkronizáló + cím + sorszám (+ stb) + hibajelző.

Tfh M adott és fix, és mi azon töprengünk, hogyan válasszuk meg N-et, hogy a legjobb legyen.

[A valóságban persze egy sokkal nagyobb N-hez kicsivel hosszabb hibajelző mező dukál.]

Következmény (①.. ⑥ feltevések együtt).

~~A pozitív nyugta = sikeres küldés valo~~

A sikeres csomagküldés = pozitív nyugta érkezésének val. sége pontosan $p := (1-p_b)^{M+N}$, és pontosan a maradék $1-p = 1-(1-p_b)^{M+N}$ val. séggel kell a csomaget újraküldeni, az előzményektől függetlenül.

A sorbanállási modelljeinket 4 lehetséges protokollra fogjuk felírni.

Megnézzük, hogy hogyen célszerű N-et megválasztani, és mennyi a csatorna kapacitása ($\frac{bit}{\mu s}$-ban).

[Emlékeztető: a példa kedvéért 1 bit elküldésének időigénye $t_0 = 1\mu s$]

$\boxed{1}$ Késleltetésmentes nyugta

Az $M+N$ bitből álló csomaget elküldjük $(M+N)\mu s$ idő alatt, majd $\boxed{\text{azonnal}}$ megérkezik a nyugta, és azonnal kiderül, hogy újra kell-e küldeni.

Az Természetes diszkrét időben nézni

1 diszkrét időegység $:= (M+N)\mu s$,

vagyis $X_n :=$ az elküldésre váró csomagok száma $n(M+N)\mu s$ elteltével.

- Az n-edik időegységben ~~elkü~~ sikeresen elküldött csomagok száma ~~$V_n \sim B(p)$~~ 0 vagy 1 lehet, konkrétan

$$V_n \sim B(p)$$

- Avagy: A k-adik csomag átjuttatásához szükséges diszkrét idő $S_k \sim Geom(p)$.

A sort ~~mode~~ leírhatjuk az $X_{n+1} = (X_n - V_{n+1})_+ + Y_{n+1}$ sorhossz-evolúciós modellel, ahol Y_n az n-edik diszkrét időrésben érkező csomagok száma,

Vagy ~~a~~ a $W_{n+1} = (W_n + S_n - T_{n+1})_+$ várakozási idő-evolúciós modellel ~~lehet T_{n+1} az~~, ahol T_n az n-edik csomag ~~érkezési~~ (diszkrét) előző érkezéstől számítva, érkezési ideje az ~~$(n-1)$~~ attól függően, hogy a ~~B~~ csomagok érkezését hogy van kedvünk modellezni.

[Láttuk: Ha $Y_n \sim B(q) \iff T_n \sim Geom(q)$, akkor a két modell ekvivalens, ilyenkor X_n és W_n is Markov lánc.]

A stabilitás feltétele $\mathbb{E}Y < p$ illetve $\mathbb{E}T > \frac{1}{p}$, mindkét esetben azt jelenti, hogy 1 diszkrét időegység alatt átlagosan legfeljebb p csomagot, vagyis N_p hasznos adatbitet tudunk át küldeni (hát persze).

$\Rightarrow$ De egy diszkrét időegység $T = (M+N)\,\mu s$ folytonos idő, $\Longrightarrow$ a csatorna kapacitása $\frac{bit}{\mu s}$-ban

$$\frac{N_p}{T} = \frac{N(1-p_b)^{M+N}}{M+N} \quad \frac{1}{\mu s}.$$

Optimalizálási feladat: Adott p_b és M mellett hogyan válasszuk N-et, hogy ez <u>max.</u> legyen?

[Nyilván: Ha N kicsi, akkor az idő nagy részét metaadat küldésével töltjük.
Ha N túl nagy, akkor gyakori lesz a hiba, és sektor kell ismételni.]

Legyen $x := \frac{N}{M}$ $\quad \alpha := -\ln(1-p_b) > 0$,

így a kapacitás

$$r := \frac{N(1-p_b)^{M+N}}{M+N} = \frac{x\,e^{-\alpha M(1+x)}}{1+x} = e^{-\alpha M}\,\frac{x\,e^{-\alpha M x}}{1+x} = \max$$

$$\Updownarrow$$

$$f(x) := \frac{x\,e^{-\alpha M x}}{1+x} = \max$$

A maximumhely kereséséhez

$$0 := f'(x) = \frac{\left(e^{-\alpha M x} - \alpha M x\, e^{-\alpha M x}\right)(1+x) - x\, e^{-\alpha M x}}{(1+x)^2} = \frac{e^{-\alpha M x}}{(1+x)^2}\left[(1-\alpha M x)(1+x) - x\right]$$

$$= \frac{e^{-\alpha M x}}{(1+x)^2}\left[1 - \alpha M x - \alpha M x^2\right] = -\alpha M \frac{e^{-\alpha M x}}{(1+x)^2}\left[x^2 + x - \frac{1}{\alpha M}\right]$$

amiből a maximumhely $x^* = -\frac{1}{2} \pm \sqrt{\frac{1}{4} + \frac{1}{\alpha M}}$,

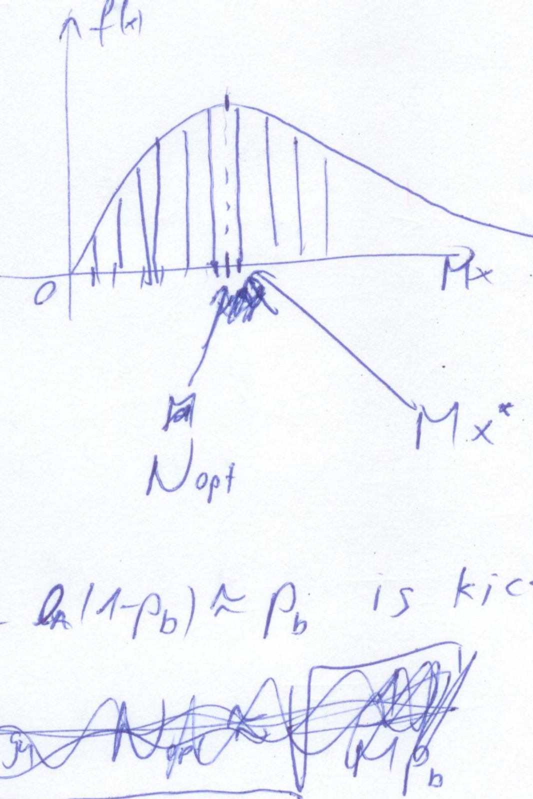
persze $x^* > 0$, vagyis $x^* = -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{\alpha M}}$.

Az optimális N érték

$$\boxed{N_{opt} = M x^* = -\frac{M}{2} + \sqrt{\frac{M^2}{4} + \frac{M}{\alpha}}}$$

Pontosabban: az ehhez legközelebbi egész számok egyike.

$f(0) = f(\infty) = 0$,
így x_* tényleg maximumhely.



Ha p_b kicsi, akkor $\alpha = -\ln(1-p_b) \approx p_b$ is kicsi

$\cancel{\Rightarrow \frac{1}{\alpha M} \gg \frac{1}{4}, \text{ így } N_{opt} \approx \sqrt{\alpha M p_b}}$

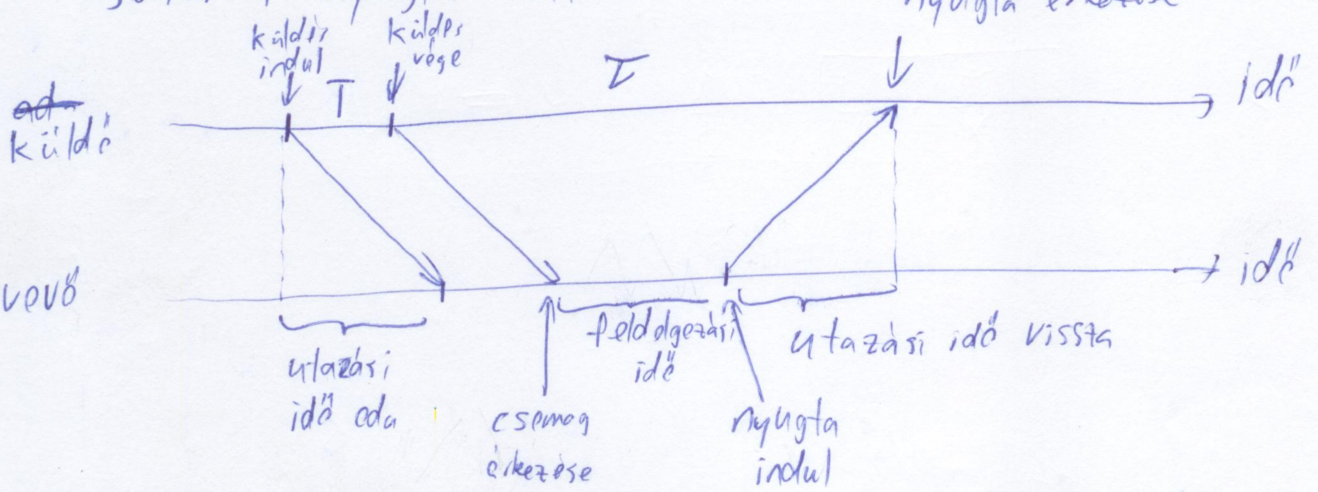
$\Rightarrow \frac{1}{\alpha M} \gg \frac{1}{4}$, így $\boxed{N_{opt} \approx \sqrt{\frac{M}{p_b}}} \rightarrow \boxed{\Gamma_{opt} \approx 1 - 2\sqrt{M p_b}}$

P1:

M	p_b	~~N_{opt}~~	r_{opt}	N_{opt}
48	10^{-5}	~~[illegible]~~	0.957	2167
48	10^{-7}	~~[illegible]~~	0.9956	21885

2 Stop-and-Wait protokoll

A vevő messze van és/vagy idő kell neki a csomag ellenőrzéséhez + nyugta küldéséhez:



ezért a csomag elküldése után megállunk, és ~~tétlenül~~ megvárjuk a nyugtát.

Pontosabban: várjuk a nyugtát, és ha τ ideig nem jön, akkor negatívnak vesszük.

$\boxed{\text{Egyszerűsítő feltevés}}$ (7): A küldés után pontosan τ időt várunk, és akkor küldjük a következő csomagot, vagy az előzőt újra.

Nyilvánvaló:

A diszkrét idejű sorbanállási modell ugyanaz, mint a késleltetés-mentes esetben volt, csak ezúttal az idő egység $T=(M+N)\,\mu s$ helyett ~~T = τ~~ $T' = T + \tau$

A kapacitás pedig $\frac{Np}{T}$ helyett $\frac{Np}{T+\tau}$.

A számolás kedvéért legyen $\tau = K\,\mu s$ (ha K nem egész, nem nagy baj),

így
$$r = \frac{N(1-p_b)^{M+N}}{M+N+K} \quad \frac{x := \frac{N}{M}}{\alpha := -\ln(1-p_b)} \quad \frac{x\,e^{-\alpha M(1+x)}}{x+y}$$
$$y := \frac{K+M}{M}$$

8-as számú egyszerűsítő feltevés: τ független ~~M-től~~ N-től, vagyis egy hosszabb csomag feldolgozására sem kell többet várni.

[Ez is életszerű: pl a CRC-t lehet valós időben számolni, mialatt a csomag érkezik.]

$\Rightarrow$ az optimális csatorna-kihasználáshoz

$$r = e^{-\alpha M}\,\frac{x\,e^{-\alpha M x}}{x+y} =: e^{-\alpha M} f(x)$$ -et kell

maximalizálnunk adott M, α és y mellett.

Az optimalizálás úgy megy, mint az előbb:

$$f'(x) = \ldots = -\alpha M \frac{e^{-\alpha M}}{(x+y)^2}\left[x^2 + yx - \frac{y}{\alpha M}\right] =: 0$$

amiből $x^* = -\frac{y}{2} + \sqrt{\frac{y^2}{4} + \frac{y}{\alpha M}}$

$$N_{opt} = -\frac{M+K}{2} + \sqrt{\frac{(M+K)^2}{4} + \frac{M+K}{\alpha M}}$$

$y = 1$, $K = 0$-ra persze visszakapjuk a késleltetésmentes esetet.

Pl:

M	p_b	K	N_{opt}	r_{opt}
48	10^{-5}	0	2167	0.957
48	10^{-5}	960	9561	0.822

Ha $M = 48$, $p = 10^{-5}$, $K = 960$ de $N_{opt} \neq N = 2167$, akkor $r = 0.668$

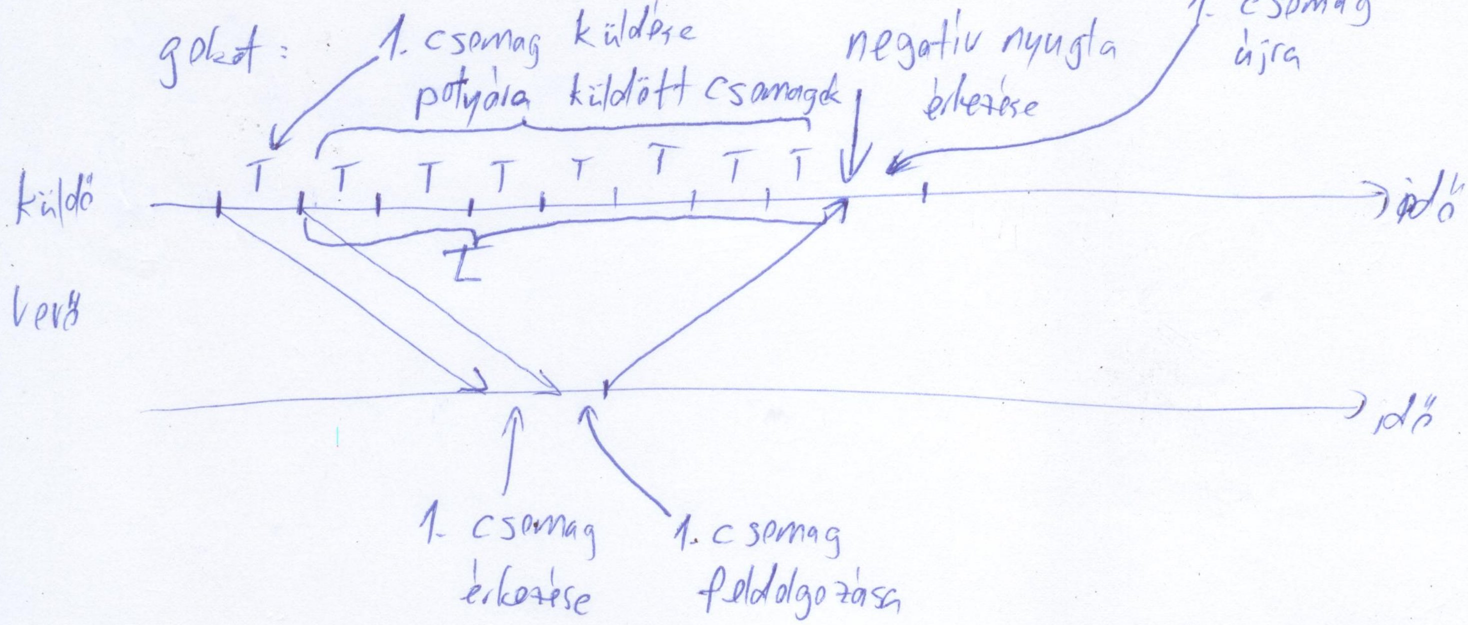
A késleltetett esetben érdemes nagyobb csomagokat küldeni.

3 Go-Back-N protokoll

Ez az „N" nem ez az „N" ☹ bocs

Most is N+M bitet küldünk el $(N+M)t_0 = (N+M)\mu s$ alatt, és most is pontosan $T = K \mu s$-ot várunk a nyugtára, ám addig se tétlenkedünk, küldjük a további csomagokat sorban.

Ha T idő múlva negatív nyugta jön (vagy semmi), akkor visszaugrunk a hibás csomaghoz, és onnan folytatjuk, sorban küldjük újra a csomagokat:



A vevő nem elég okos ahhoz, hogy az összekeveredett csomagokat sorbarakja — lehet, hogy a hiba észlelése után már nem is figyel a többi csomagra, csak a hibásat várja újra.

Észrevétel: Nem érdekes, hogy mi lett (volna) a petyára küldött csomagokkal.

Hanem: Amikor az n-edik csomagot ~~helyesen~~ $\boxed{\text{hibátlanul}}$ elküldtük, akkor kezdődik érdemben az $(n+1)$-edik küldése

bár ez csak később fog kiderülni.

Köv: Az átvitel $\boxed{\text{FIFO}}$, a (folytonos) átviteli

idő $$S_n^{\text{folyt}} = \begin{cases} T, & \text{ha 1.-re sikerül} \\ T + \tau + T, & \text{ha 2.-ra sikerül} \\ \vdots & \\ T + (\tau + T)\cancel{(k)}(k-1), & \text{ha } \cancel{k\text{-adikra}} \text{ sikerül} \\ & k\text{-adikra} \end{cases}$$

$k = 1, 2, 3 \dots$

A $W_{n+1} = (W_n + S_n - T_{n+1})_+$ késleltetés-evolúciós egyenlet továbbra is érvényes (diszkrét időben is, ha $\tau = K\ \mu s$, és a diszkrét időegységnek $1\ \mu s$-ot választjuk.)

Legyen $U_n \in \{1, 2, 3 \dots\}$ az, hogy hanyadik próbálkozásra sikerül átküldeni az n-edik csomagot.

Ekkor $U_n \sim Geom(p)$ ahol továbbra is $p = (1-p_b)^{M+N}$

és $S_n = (M+N) + (M+N+K)(U-1)$, (diszkrét átviteli idő, μs-ben) amiből $\mathbb{E}S_n = (M+N) + (M+N+K)(\mathbb{E}U - 1) =$

$$= (M+N) + (M+N+K)\left(\frac{1}{p} - 1\right) = \frac{M+N+K}{p} - K$$

A stabilitás feltétele $\mathbb{E}T_n > \mathbb{E}S_n$, ami most is azt jelenti, hogy diszkrét időegységenként (vagyis μs-onként) átlagosan legfeljebb $\frac{1}{\mathbb{E}S_n} = \frac{p}{M+N+K(1-p)}$ db csomagot,

azaz $r = \frac{N}{\mathbb{E}S_n} = \frac{Np}{M+N+K(1-p)}$ db adatbitet tudunk átküldeni.

Egyben: A csatorna kapacitása

$$\boxed{r = \frac{N(1-p_b)^{M+N}}{M+N+K\left[1-(1-p_b)^{M+N}\right]}}$$

[$K=0$-ra persze visszakapjuk a késleltetés-mentes esetet.]

Ennek az optimalizálása sajnos csak numerikusan megy.

Megj: Az $X_{n+1} = (X_n - V_{n+1})_+ + Y_{n+1}$ sorhossz-modell NEM illik rá a Go-Back-N protokollra, mert az egymást követő ~~sikeres~~ időrésekben sikeresen átküldött csomagok száma nem lesz független, akárhogy választunk is időegységet: Minden kudarcot üresjáratok követnek. !!!

$\boxed{4}$ Szelektív ismétlés

Mint a Go-Back-N protokollnál: folyamatosan küldjük a csomagokat, mindegyiket $(N+M)$ μs alatt, majd K μs-et várunk a nyugtára, de

- Csak azt küldjük újra, amelyiknél hiba történt.
- A vevő ügyes, és addig is gyűjti a helyesen érkező csomagokat, amíg a hibásat újra várja.
- A vevő okos, van tárhelye, és majd a végén sorbarakja az össze-vissza sorrendben beérkező csomagokat.

Észrevétel 1: Ez $\boxed{\text{NEM}}$ FIFO rendszer, ezért a ~~sor~~ várakozási idő-evolúciós modell nem illik rá.

Észrevétel 2: A sorhossz viszont Markov, az

$$X_{n+1} = (X_n - V_{n+1})_+ + Y_{n+1}$$

modell érvényes, és ezzel a rendszer $\boxed{\text{UGYANAZ,}}$ mint a késleltetés mentes nyugta esete.