

Az 1998. októberi számunkban a Forgalm szabályozás az Interneten című írásban *Éltető Tamás*, az **Ericsson Traffic Lab** munkatársa a TCP protokoll működését vázolta fel. A cikk végén egy feladatot tűzött ki, amelyben azt kellett meghatározni, hogy a leírt modell szerint működő rendszerben

- (a) mennyi idő alatt lehet 800 000 bit információt zavartalanul továbbítani;
- (b) hosszú távon mekkora az átlagos átviteli sebesség.

A rendszer paraméterei – maximális sebesség: 26400 bit/sec – arányossági tényező a felderítési szakaszban: 0,6931 1/sec; az üzenetküldési szakaszban: 1200 bit/sec².

A pályázatra két dolgozat érkezett. *Máthé András* (Budapest, ELTE, Apáczai Csere J. Gyak. Gimn.) tömör megoldásában kiszámította az eredményt. *Szóllósi Loránd* (Hajdúszoboszló, Hőgyes E. Gimn.) pedig megadta a helyes képleteket.

Mindketten egy-egy **Ericsson 128TA** ISDN modemet kaptak ajándékkul az ERICSSON Távközlési Kft-től. Az Ericsson 128TA adapterrel 2 × 64 Kbit/s adatátviteli sebességgel lehet internetezni ISDN vonalon. Emellett hagyományos telefonkészülék is kapcsolható hozzá, és a számítógép soros kimenetéhez kötve modemként használható.

Most pedig nézzük a probléma megoldását.

A feladat szerint az adattovábbítás két fő szakaszból áll: a felderítési, és az üzenetküldési szakaszból. A felderítési szakaszban és az üzenetküldési szakasz elején egy rövid ideig exponenciálisan nő a sebesség. Az üzenetküldési szakasz további részében lineáris periódusok követik egymást, ahol a sebesség lineárisan nő $\frac{s_{\max}}{2}$ -ről egészen $s_{\max}$ -ig. Bevezetünk néhány jelölést:

s_{1,2}(t) Az adattovábbítás sebessége a felderítési szakaszban, illetve az adatküldési szakasz elején, illetve az üzenetküldési szakaszban.

s* A kezdősebesség a felderítési szakasz elején.

a Az arányossági tényező a felderítési szakaszban.

b Az arányossági tényező az üzenetküldési szakaszban.

s_{max} A csatornán elérhető maximális sebesség.

t_{max1,2} A felderítési szakasz időtartama, illetve az üzenetküldési szakasz egy periódusának (lineáris sebességnövekedés) időtartama.

t_{koz} Az üzenetküldési szakasz elején, az exponenciális sebességnövekedés időtartama.

Δt Az üzenetküldés végén fennmaradt csonka periódus időtartama.

D_{1,2,3} Az elküldött adat mennyisége a felderítési szakaszban, az üzenetküldési szakasz elején, és az üzenetküldési szakasz egy periódusában.

D_{Δt} Az elküldött adat mennyisége az üzenetküldési szakasz végén a csonka periódusban.

T Az üzenet elküldésének teljes ideje.

Először írjuk fel a sebességet a felderítési szakaszban! $s_1'(t) = as_1(t)$, $s_1(0) = s^*$ -ből kiindulva, a differenciálegyenletet megoldva:

$$s_1(t) = s^* \cdot e^{at}.$$

Most írjuk fel a sebességet az üzenetküldési szakasz egy periódusában.

$$s_2(t) = bt + d, \quad s_2(0) = \frac{s_{\max}}{2} = d,$$

$$s_2(t) = bt + \frac{s_{\max}}{2}.$$

A két sebesség-függvény segítségével írjuk fel a különböző szakaszok időtartamát!

t_{max1}:

$$s_1(t_{\max1}) = s^* \cdot e^{at_{\max1}} = s_{\max}, \quad \text{innen} \quad t_{\max1} = \frac{\ln \frac{s_{\max}}{s^*}}{a}.$$

t_{koz}:

$$s_1(t_{\text{koz}}) = \frac{s_{\max}}{2}, \quad \text{innen} \quad t_{\text{koz}} = \frac{\ln \frac{s_{\max}}{2s^*}}{a}.$$

t_{max2}:

$$s_2(t_{\max2}) = bt_{\max2} + \frac{s_{\max}}{2} = s_{\max}, \quad \text{innen} \quad t_{\max2} = \frac{s_{\max}}{2b}.$$

Mivel felírtuk mind a sebességeket, mint az időtartamokat, ezekből fel lehet írni az átvitt adatmennyiségeket a különböző szakaszokban. Némi számolással

$$D_1 = \int_0^{t_{\max1}} s_1(u) du = \frac{s_{\max} - s^*}{a}, \quad D_2 = \int_0^{t_{\text{koz}}} s_1(u) du = \frac{s_{\max} - 2s^*}{2a},$$

$$D_3 = \int_0^{t_{\max2}} s_2(u) du = \frac{3s_{\max}^2}{8b}.$$

$$D_{\Delta t} = \int_0^{\Delta t} s_2(u) du = \frac{b}{2}(\Delta t)^2 + \frac{s_{\max}}{2}\Delta t, \quad D_{\Delta t} < D_3, \quad \Delta t > 0.$$

Δt -t eddig nem tudtuk felírni, mert ismerni kell hozzá $D_{\Delta t}$ -t, amit mindjárt meg is kapunk. Az adatátvitel teljes ideje, és a teljes átvitt adatmennyiség egyszerre kapható:

$$T = t_{\max 1} + t_{\text{koz}} + kt_{\max 2} + t_{\Delta t}, \quad D = D_1 + D_2 + kD_3 + D_{\Delta t},$$

ahol

$$k = \left\lceil \frac{D - (D_1 + D_2)}{D_3} \right\rceil.$$

Így itt az üzenetküldési szakasz k darab egész periódusból áll. Ekkor $D_{\Delta t}$ nyilván:

$$D_{\Delta t} = D - (D_1 + D_2 + kD_3),$$

és Δt pedig

$$\frac{b}{2}(\Delta t)^2 + \frac{s_{\max}}{2}\Delta t - D_{\Delta t} = 0.$$

Ebből

$$\Delta t = \frac{\sqrt{s_{\max}^2 + 8bD_{\Delta t}} - s_{\max}}{2b}.$$

A feladatban nem volt megadva s^* értéke, ami nélkül pontosan nem lehet kiszámítani T értékét. Közelítőleg azonban kiszámíthatjuk, ha elhanyagoljuk D_1 -et, és D_2 -t (hosszabb adatátvitelnél ez a rész nem jelentős):

$$D_1 \approx 0, \quad D_2 \approx 0, \quad D \approx kD_3 + D_{\Delta t},$$

$$k \approx \left\lceil \frac{D}{D_3} \right\rceil, \quad T \approx kt_{\max 2} + \Delta t.$$

Behelyettesítve az ismert értékeket:

$$t_{\max 2} = \frac{26400 \frac{\text{bit}}{\text{s}}}{2 \cdot 12000 \frac{\text{bit}}{\text{s}^2}} = 1.1 \text{ s}, \quad D_3 = \frac{3 \cdot 26400^2 \frac{\text{bit}^2}{\text{s}^2}}{8 \cdot 12000 \frac{\text{bit}}{\text{s}^2}} = 21780 \text{ bit}.$$

$$k \approx \left\lceil \frac{800000 \text{ bit}}{21780 \text{ bit}} \right\rceil = 36, \quad D_{\Delta t} \approx 800000 \text{ bit} - 36 \cdot 21780 \text{ bit} = 15920 \text{ bit}.$$

$$\Delta t \approx \frac{\sqrt{26400^2 \frac{\text{bit}^2}{\text{s}^2} + 8 \cdot 12000 \frac{\text{bit}}{\text{s}^2} \cdot 15920 \text{ bit}} - 26400 \frac{\text{bit}}{\text{s}}}{2 \cdot 12000 \frac{\text{bit}}{\text{s}^2}} = 0.9 \text{ s}.$$

$$T \approx 36 \cdot 1.1 \text{ s} + 0.9 \text{ s} = 39.6 \text{ s} + 0.9 \text{ s} = 40.5 \text{ s}.$$

A hosszú távon vett átlagos átviteli sebességet viszont pontosan ki tudjuk számolni. Ahogy nő az adatmennyiség, és ezzel együtt az átviteli idő, úgy nő az üzenetküldési szakasz periódusainak aránya az összes adatmennyiségben. Ezért a hosszú távon vett átlagos átviteli sebesség körülbelül egyenlő az egy perióduson belül vett átlagos átviteli sebességgel:

$$\bar{s} = \frac{\frac{s_{\max}}{2} + s_{\max}}{2} = \frac{3s_{\max}}{4} = \frac{3 \cdot 26400 \frac{\text{bit}}{\text{s}}}{4} = 19800 \frac{\text{bit}}{\text{s}}.$$

★

Ez a feladat a TCP (Transmission Control Protocol) működésének lényegét modellezte, de az Internet meghatározó jellegétől – attól, hogy ez egy csomagkapcsolt hálózat – elvonatkoztatva. Most úgy közelítjük meg a TCP működését, hogy megnézzük, mi is történik csomagszinten. Ez a megközelítés igazából még most sem a valóságos helyzet.

Mint már említettük, a *feladó* az üzenetet csomagokra bontja, és azokat külön-külön küldi el a *vevőnek*. A *vevő* minden egyes csomagot nyugtáz, mégpedig úgy, hogy visszaküld a *feladónak* egy csomagot, amelyben jelzi, hogy az üzenet elejétől kezdve hány csomagot kapott meg eddig. (Azaz, hogy mi a sorszáma a legutolsó olyan csomagnak, ameddig még valamennyi csomag hiánytalanul megérkezett.)

Azzal az optimista egyszerűsítő feltétellel élünk, hogy két, egymás után küldött csomag nem vész el, sőt, a csomagvesztések között „aránylag hosszú” idő telik el. Előfordulhat viszont a csomagvesztésen kívül másfajta hiba is: két csomag felcserélődhet a hálózaton, és a *vevő* egy később feladott csomagot kap meg korábban. Ilyenkor mindaddig nem tudja nyugtázni a kapott csomag sorszáma, amíg meg nem érkezik a hálózatban bolyongó korábbi csomag. A protokoll szerint ezt a hibát a *vevő* úgy jelzi, hogy a legutolsó, rendben megkapott csomagról küldött nyugtázó üzenetet küldi el ismét. Az ilyen üzenetet *ismételt nyugtának* nevezik.

Mit tesz eközben a *feladó*? Mint várható, számon tartja azokat a csomagokat – pontosabban ezek sorszáma – , amelyeket már elküldött, de még nem érkezett rájuk nyugtázó üzenet. Ezeket egy *torlódási ablaknak* nevezett

intervallumban tárolja. Ennek alsó végpontja az a legkisebb sorszám, amelyet a *vevő* még nem nyugtázott, a felső végpont pedig – amely, mint látni fogjuk, általában törtszám – a soron következő elküldendő csomagot jelöli ki. A *feladó* akkor küldi el a soron következő csomagot, ha a torlódási ablak felső végpontja átlépi a következő egész értéket és az ablakban még vannak el nem küldött csomagok.

A *feladónak* különböző állapotai vannak. Ezeket most soravesszük, és megnézzük, hogyan változik a torlódási ablak.

Kezdetben a *feladó* az úgynevezett *indulási állapotban* van, az alsó és a felső határ értéke 1, a torlódási ablak kijelöli az üzenet első csomagját, amelyet a *feladó* elküld. Amikor megkapja a nyugtát, 1-gyel növeli az ablak méretét a következő ármányos módon: az alsó határ 1-ről 2-re nő, a felső határ szintén, és a *feladó* elküldi a 2. csomagot. Most kerül sor az ablak méretének növelésére, a felső határ kitolódik, 3 lesz. Az ablak mérete eggyel nőtt, a következő elküldendő csomag a 3.

Az indulási állapotban minden megkapott nyugta után így nő 1-gyel az ablak mérete. A *feladó* addig marad indulási állapotban, amíg csomagvesztést nem tapasztal. Mint már írtam, ha egy csomag elvész, akkor a *vevő* a beérkező további csomagokra a legutolsó, rendben megkapott csomag nyugtáját ismétleteti. A *feladó* a második ismételt nyugta után dönt úgy, hogy baj van, elveszett az utolsó, rendben nyugtázott küldeményt követő csomag. Ha ugyanis csak egy ismételt nyugtát kap, és ezután minden helyreáll, akkor csak annyi történt, hogy az utolsó két csomag fölcserélődött.

Amikor beérkezik a harmadik azonos – tehát a második ismételt – nyugta, akkor a *feladó* állapota megváltozik. Az új állapotot *csomagvesztés utáni felépülésnek* nevezik. Ebben az állapotban még egyszer újra elküldi a *vevőnek* az elveszett csomagot, azt, amelyik az ismételt nyugtában közölt sorszám után következik.

Amíg ez a csomag meg nem érkezik a *vevőhöz*, addig az minden újabb csomag érkezésekor – amelyeket félretesz – csökkenően ismétleteti a nyugtát. Amikor megkapja a hiányzó csomagot, összerakja az üzenet félretett csomagjait, beilleszti az addig hiányzót és nyugtázza a rendben megkapott csomagok közül a legnagyobb sorszámút. A *feladó* eközben azt feltételezi, hogy csak egy csomag veszett el és az ismételt nyugták a későbbi csomagok megérkezését jelzik. Így az ismétlődő nyugtákra is 1-gyel növeli a torlódási ablak méretét. Erre azért van szükség, mert az ismételt nyugták nem növelik automatikusan az ablak felső határát, így pedig a *feladó* nem küld új csomagot, és feleslegesen várna, amíg meg nem érkezik az elveszett csomag pótlását jelző nyugta.

Amikor helyreáll a rend és beérkezik az első, nem ismételt nyugta, akkor – értelemszerűen – a *feladó* ismét állapotot vált. Az új állapotot *torlódás-elkerülési állapotnak* nevezik.

A torlódás-elkerülési állapotba lépés után a *feladó* nem növeli az ablak felső határát, és addig nem küld újabb csomagot, amíg a torlódási ablak hossza a felére nem csökken. Ha nem veszett el újabb csomag, akkor ez azonnal megtörténik, mert az elveszett csomag megkerülését éppen a legnagyobb sorszámú csomagról szóló nyugta jelzi.

Miután a torlódási ablak a felére csökkent, újra elkezd növelni az ablak felső határát és újabb csomagokat küld minden olyan esetben, amikor a felső határ átlép egy egész számot. Ettől kezdve viszont már nem egyesével nő tovább az ablak felső határa az újabb nyugták beérkezésekor, hanem lassabban. A *feladó* a következő formulát használja:

$$\text{új méret} = \text{régi méret} + \frac{1}{\text{régi méret}}.$$

Az októberben kitűzött feladat szellemében ezután a *feladó* egészen a következő csomagvesztés érzékeléséig marad torlódás-elkerülési állapotban. Ha ez bekövetkezik – és most föltesszük, hogy a torlódási ablak mérete ezúttal is pontosan akkora, mint az első csomagvesztéskor, mert csak mi használjuk a hálózatot –, akkor felépülési állapotba kerül, amiből újra torlódás-elkerülési állapotba jut.

Feltételezzük még, hogy a csomagok utazási ideje állandó, továbbá, hogy a *vevő* nyomban elküldi a beérkezett csomagok nyugtáját, azaz az épségben megérkező csomagok elküldése után a *feladó* mindig ugyanannyi idővel kapja meg a nyugtát.

A *kérdés a következő*: miért felel meg az októberi közleményben leírt modell a TCP most ismertett működésének, azaz annak, ahogyan a *feladó* az adás sebességét változtatja?

A helyes választ beküldőket ezúttal is az ERICSSON Távközlési Kft jutalmazza.

Éltető Tamás

Ericsson Traffic Lab

