BHG ORION TERTA
MŰSZAKI KÖZLEMÉNYEK
XXVI. évfolyam 1980. 7. szám

Ciklikus lefoglalású memóriaterületek forgalmi méretezése tárolt programvezérlésű telefonközpontban

CSÁSZÁR GYULA
SCHULLER J. ATTILA
BHG

Bevezetés

Tárolt programvezérlésű (TPV) rendszerekben a vezérlő program munkája során mind hagyományos telefonközponti, mind egyéb feladatokhoz különböző memória területeket használ fel. Ilyen memória területekre lehet szükség egyrészt pl. a különféle (kimenő-, bejövő-) hívások hívószámainak, másrészt a folyamatban levő hívásokra vonatkozó egyéb adatok tárolásához, amelyeket egyes rendszerekben a hívásokhoz rendelt, e célra kijelölt memória-részekben gyűjtenek össze. Ha egy adott memória-terület megtelik adatokkal, akkor a TPV-ben torlódás jön létre, ami várakozást vagy veszteséget eredményezhet.

A torlódás elkerülésére a memória területeket forgalmilag méretezni kell.

A memória területek egy-egy részét — amelyek pl. az egy hívástól (igénytől) származó adatok tárolásához szükségesek — kiszolgáló egységnek tekintjük és az alábbiakban hívástárnak nevezzük. A forgalmi méretezéshez az igények beérkezési sűrűségén és a hívástárra vonatkoztatott tartásidején kívül figyelembe kell venni azt is, hogy a szabad hívástárat a vezérlő programrendszer hogyan keresi meg a kérdéses memória területen belül.

A méretezendő rendszer leírása után a vizsgálat alapelvét ismertétjük, majd bemutatjuk az elkészített utánzóprogramot és a kapott matematikai összefüggéseket. Ezt követi néhány számszerű eredmény. A matematikai levezetéseket függelék tartalmazza.

Az ismertetett vizsgálatra a BHG QA típusú TPV alközpontjainak kifejlesztése kapcsán volt szükség.

1. A rendszer

Nincs elvi akadálya annak, hogy egy igény jelentkezésekor a teljes memória területet végignézzük szabad hívástár keresésekor. A vezérlő programrendszer terhelésének csökkentése érdekében azonban többnyire olyan keresési módot választanak, hogy egy-egy alkalommal csak előre meghatározott számú — k darab — hívástárat vizsgálnak meg, és ha ezek között nincs szabad, akkor torlódás jön létre. A vizsgált esetben a keresés kötött sorrendben halad végig a teljes csoporton, a következő keresési műveletsor az utolsó megállási helyzetből indul tovább (az utoljára lefoglalt hívástártól vagy a sikertelen sorozat véghelyzelétől). A csoport utolsó tagja után a keresés a csoport első tagjával folytatódik. Az eljárást k lépéses ciklikus keresésnek nevezhetjük.

A forgalmi méretezés célja esetünkben az, hogy mind a szükséges teljes memória terület nagyságát, mind pedig az előírt szolgáltatási szinthez megfelelő k lépésszámot meghatározza.

A fenti rendszerre vonatkozó matematikai eredményeket az irodalom tudomásunk szerint nem közöl. Saját, a veszteséges esetre vonatkozó vizsgálataink is csak k speciális értékeire szolgáltattak eredményt. A kérdés bonyolultságát a Függelék szemlélteti. A forgalmi méretezési feladatot végeredményben utánzással oldottuk meg. A program néhány bemutatott részletéből látható, hogy a matematikailag nehezen kezelhető problémához egyszerű utánzóprogram készíthető.

2. Matematikai modell

Adott n darab hívástár — ezeket a szemléletesség kedvéért az 1. ábrán egy körben helyeztük el, feltüntetve a mutató haladási irányát is*; funkcióját blokksémaszerűen a 2. ábra mutatja be.

Minden igény érkezésekor lezajlik egy keresési eljárás, melynek eredménye, hogy a mutató legalább egy, de legfeljebb $k(<n)$ lépést tesz meg. Ha a k lépéses keresés sikertelen, akkor az igény elvész.

Feltesszük, hogy az igények érkezése Poisson-folyamatot alkot, és a kiszolgáló egységek foglaltsági ideje exponenciális eloszlású. A szükséges műveletek (keresés, szabad hívástár lefoglalása, stb.) idejét

* A mutató a sorrendben következő hívástárat jelöli ki.

elhanyagoljuk, és ily módon egy speciális kiszolgálású, (M/M/n típusú) tömegkiszolgálási rendszert kapunk.

A feladat az előbb leírt modell jellemzőinek a meghatározása, a statisztikai egyensúly állapotában.

Jelölje λ a Poisson-folyamat, μ pedig az exponenciális eloszlású kiszolgálási idő paraméterét.

Jelölje q_s annak a valószínűségét, hogy éppen s db hívástár foglalt ($s=0, 1, \ldots, n$), valamint jelölje $V_k(n)$ a keresett veszteségi valószínűséget, amely az $A=\frac{\lambda}{\mu}$-nek, azaz a felajánlott forgalomnak lesz a függvénye.

Az említett jellemzőket meghatározni csupán az alábbi két esetben sikerült:

A) $k=1$ $\quad V_1(n)=\left(\frac{A}{A+1}\right)^n$,

$$q_{n-j}=\frac{A^n}{\prod\limits_{i=1}^{n-j}(A+i)}\sum_{l_1=1}^{n-j+1}\frac{l_1}{(A+l_1-1)(A+l_1)}\sum_{l_2=1}^{l_1+1}\cdots\sum_{l_j=1}^{l_{j-1}+1}\frac{l_j}{(A+l_j-1)(A+l_j)}.$$

Ebben a kifejezésben j db összegezés szerepel ($j=1, \ldots, n$), $j=0$-ra pedig csak az első tényező marad:

$$q_n=\frac{A^n}{\prod\limits_{i=1}^{n}(A+i)}.$$

B) $k=n-1$

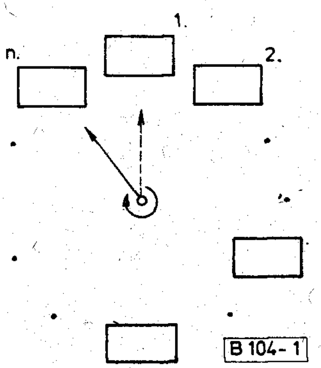
$$V_{n-1}(n)=\frac{\frac{A^n}{n!}(A+n)}{(A+n)\sum\limits_{i=0}^{n}\frac{A^i}{i!}-\frac{1}{n}\sum\limits_{i=0}^{n}\frac{A^i}{i!}(n-i)},$$

$$q_0=\left\{\sum_{i=0}^{n-1}\frac{A^i}{i!}+\frac{A^n}{(n-1)!}\,\frac{A+n-1}{(n+1)\cdot A+n(n-1)}\right\}^{-1},$$

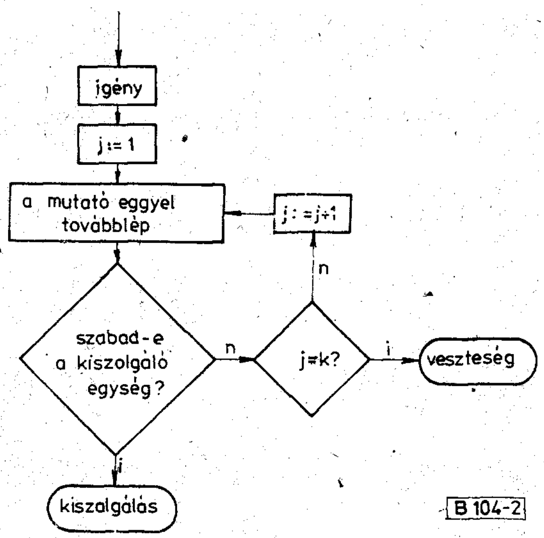
$$q_s=\frac{A^s}{s!}q_0 \quad 0\le s<n\text{-re, és}$$

$$q_n=\frac{A^n}{(n-1)!}\,\frac{A+n-1}{(n+1)\cdot A+n(n-1)}q_0.$$

Ezen összefüggések levezetését a függelékben írjuk le.



1. ábra. A ciklikus keresési elv szemléltetése



2. ábra. A k lépéses ciklikus kiszolgálási elv

3. Számítógépes utánzás

A rendszer teljes vizsgálatához utánzóprogramot készítettünk, amely az MTA SzTAKI CDC 3300-as számítógépén futtatható. A program SIMULA nyelven íródott. A nyelv általános rendszerleíró eszközeivel teljesen hűen megfogalmazható és utánozható a hívástárak ciklikus lefoglalása.

A nyelv legfontosabb fogalma az osztály. Ez az ALGOL-eljárásokhoz hasonlóan deklarálható minta, és új típusú blokkpéldányok, úgynevezett objektumok létrehozását teszi lehetővé. Az igen lényeges különbség az, hogy az adatstruktúra és az eljárás hivatkozások az objektum törzsében szereplő utasítások végrehajtása után is rendelkezésre állnak. Ugyanannak az osztálytörzsnek tetszőleges számú példánya létezhet egyidejűleg, mindegyikre nevekkel hivatkozhatunk; mindegyik példány lokális változóit elérhetjük (lásd pl. [3]).

Programunknak két fő alkatrésze van, ezek a HÍVÁSTÁR, illetve HÍVÁS nevű osztályok.

A hívástár tevékenységeit a következőképpen foglaltuk össze:

```
PROCESS CLASS HÍVÁSTÁR;
  BEGIN BOOLEAN FOGLALT;
  WHILE TRUE DO BEGIN
    FOGLALT:=TRUE;
    ...;
    HOLD (amíg foglalt a hívástár);
    FOGLALT:=FALSE;
    ...;
    PASSIVATE;
                        END;
    END;
```

Tehát a HÍVÁSTÁR osztályba tartozó objektumok lényege, hogy lefoglalódnak és bizonyos idő után újra szabaddá válnak. Ez az időtartam — ami esetünkben exponenciális eloszlású valószínűségi változó — a paramétere a HOLD (.) utasításnak, melynek hatására az időben következő folyamat (PROCESS) végzi el a szükséges teendőket.

A hívások érkezését a következő osztály írja le:

```
PROCESS CLASS HÍVÁS;
 BEGIN BOOLEAN PROCEDURE
 TALÁL;...;
  WHILE hívásszám ≤ maximum DO
   BEGIN ...;
    IF NOT TALÁL THEN veszteség
     ELSE ACTIVATE MUTATÓ;
     HOLD (szünetidő);
   END;
 END;
```

A „TALÁL" eljárás értelemszerűen akkor *TRUE*, ha legfeljebb k lépésben talált szabad hívástárat a hívás, és az *ACTIVATE* utasítás hatására az a HÍVÁSTÁR-objektum kapja a vezérlést, amelyre a MUTATÓ utal. Miután az lefoglalódott, a HÍVÁS-objektum kapja vissza a vezérlést, és a HOLD (.) utasítás hatására újabb idő eltelte után aktivizálódik.

A vezérlésátadó HOLD, PASSIVATE, *ACTIVATE* utasítások következtében az egy HÍVÁS, és az n darab HÍVÁSTÁR-objektum mintegy egymással párhuzamosan működik a kívánt módon, időhű utánzást valósítva meg.

Az elmondottak a veszteséges esetre vonatkoznak. Várakozásos ciklikus rendszer utánzása néhány kiegészítést igényel, de nem jelent nehézséget.

4. Az utánzás eredményei

Az elkészült programmal vizsgálatokat végeztünk egy 98 hívástárát tartalmazó veszteséges rendszerrel kapcsolatban. 5×5000 hívásból álló sorozatokat vizsgáltunk „bemelegítés" után, 68 Erlang felajánlott forgalom mellett, különböző k értékekre, illetve $k=6$ esetén különböző felajánlott forgalmakra. Az alábbi táblázatok a kapott eredményekkel együtt tartalmazzák a 95%-os megbízhatósági szintnél számított konfidencia intervallumokat is.

k	Utánzási eredmények (n = 98, A = 68 erl.)	
	Veszteség aránya [%]	Átvitt forgalom [erl]
5	2,66 (1,92, 3,40)	67,184 (65,656, 68,712)
6	1,87 (1,40, 2,34)	67,784 (66,081, 69,487)
7	1,26 (0,90, 1,62)	67,480 (67,000, 67,963)
8	0,84 (0,51, 1,17)	68,053 (66,628, 69,478)

A [erl]	Utánzási eredmények (n = 98, k = 6)	
	Veszteség aránya [%]	Átvitt forgalom [erl]
62	0,50 (0,39, 0,61)	61,656 (60,641, 62,671)
68	1,87 (1,40, 2,34)	67,784 (66,081, 69,487)
75	3,92 (3,04, 4,80)	72,792 (71,575, 74,009)

Az eredmények megfelelnek a rendszerről alkotott általános elképzeléseinknek (pl. a k növekedésével csökken a veszteség).

Ezenkívül leszűrhetjük azt, hogy adott szolgáltatási szint (pl. 1% alatti veszteség) elérése után már nem érdemes a k lépésszámot növelni, mivel a jellemzők javulásának mértéke egyre csökken, viszont a keresési eljárás ideje hosszabbodik. Ez főleg kisebb nyalábok esetén igaz, amint azt az n=20-ra vonatkozó alábbi táblázatok mutátják:

k	n = 20, A = 10 [erl]	
	Veszteség aránya [%]	Átvitt forgalom [erl]
3	2,43 (2,23, 2,64)	9,761 (9,544, 9,977)
4	1,43 (1,01, 1,85)	9,853 (9,814, 9,892)
5	0,72 (0,51, 0,93)	9,927 (9,883, 9,971)
6	0,67 (0,37, 0,98)	10,077 (9,738, 10,416)

A [erl]	n = 20, k = 5	
	Veszteség aránya [%]	Átvitt forgalom [erl]
10	0,72 (0,51, 0,93)	9,927 (9,883, 9,971)
11	1,56 (1,33, 1,80)	10,927 (10,630, 11,222)
12	2,67 (1,96, 3,37)	11,682 (11,384, 11,98)

Ismeretes, hogy teljes elérhetőségű rendszerek esetében ($k=$n) a kisebb méretű nyalábok kevésbé érzékenyek a túlterhelésre. (Ugyanolyan arányú forgalomnövekedés nem okoz akkora veszteség-növekedést — az adott szolgáltatási szint környezetében.)

Ez a jelenség megfigyelhető most is. Emellett még arra is következtethetünk, hogy rögzített nyalábméret esetén a ciklikus keresés k lépésszámának növekedésével együtt növekszik a rendszer érzékenysége. Ez is amellett szól, hogy a lépésszámot ne növeljük tovább, ha már teljesült az előírt szolgáltatási szint.

Függelék

Számozzuk meg a hívástárakat 0-tól (n−1)-ig, és jelölje t_i az i-ik állapotát, mely 0, ha az szabad, és 1, ha foglalt. Valamint legyen $u=0, 1, \ldots,$ n−1 annak a tárnak a sorszáma, amelynél a mutató áll — ezzel mod(n) kell számolni.

Az $(l_0, l_1, \ldots, l_u, \ldots, l_{n-1};\ u)$ vektor leírja az egész rendszer állapotát — de amint az meggondolható, stacionárius esetben az állapotok valószínűségei attól függnek, hogy a mutató előtt milyen sorrendben vannak a szabad, illetve foglalt hívástárak, viszont független az u sorszámtól. Tehát elegendő az $L=(l_{u+1}, l_{u+2}, \ldots, l_{u+n-1}, l_u)$ vektorokat tekinteni, és ha $p(L)$ jelöli ennek valószínűségét, akkor

$$\sum_L p(L)=\frac{1}{n}. \qquad (1)$$

A könnyebb kezelhetőség érdekében célszerű megszámozni az ismeretlen $p(L)$ valószínűségeket, mégpedig a következőképpen:

$$p(L)=p(l_{u+1}, \ldots, l_{u+n-1}, l_u) \leftrightarrow X_{m(L)}, \quad \text{ahol}$$

$$m(L):=l_{u+1}\cdot 2^{n-1}+l_{u+2}\cdot 2^{n-2}+\ldots+l_u. \qquad (2)$$

Ily módon 0-tól (2^n-1)-ig számozott, 2^n db. ismeretlenünk van, és (1) szerint

$$\sum_{i=0}^{2^n-1} X_i=\frac{1}{n}. \qquad (1')$$

Az állapotvalószínűségekre a következőképpen írhatók fel egyenletek: (Lásd pl. [1]; 59. o.)

Minden állapotra felírható az az arány, amely szerint a valószínűség „befolyik", illetve „távozik" ebből az állapotból, és e két mennyiség egyenlő. Mivel ezen állapotegyenletekre nem lesz szükségünk részletesen, csak az alábbi esetre írjuk fel őket.

Tekintsük az X_{2^n-2}-t, vagyis az $L=(1, 1, \ldots, 1, 0)$ állapot valószínűségét. Ez az az állapot, melyben az összes kiszolgáló foglalt — egy kivétellel, és éppen erre a szabad kiszolgálóra hivatkozik a mutató. Ebből az állapotból kimenet kétféleképpen történhet:

kiszolgálás megszűnésével: aránya: $(n-1)\cdot\mu\cdot p(L)$
(μ a tartásidő paramétere);
új igény érkezésével: aránya: $\lambda\cdot p(L)$
(λ a Poisson folyamat paramétere).

Az állapotba viszont csak egyféleképpen lehet „bemenni":

kiszolgálás megszűnésével: $\mu\cdot p(1, 1, \ldots, 1)$.

Új igény érkezésével azért nem juthatunk ebbe az állapotba, mert amint az a 2. ábrából is kiderül, a mutató mindig foglalt kiszolgálóra mutat ilyen esetben. Tehát az L-re vonatkozó egyenlet:

$$(A+n-1)X_{2^n-2}=X_{2^n-1}. \qquad (3)$$

Az egyenlet itt már μ-vel osztott alakban szerepel, és $A=\lambda/\mu$ a felajánlott forgalom.

Általában a felírható állapotegyenletek az ismeretleneknek csak az arányát határozzák meg egyértelműen, és azokhoz még hozzá kell venni azt a feltételt, hogy az összes állapot valószínűségének összege 1. Jelen esetben ez az (1)-es összefüggéssel ekvivalens.

Az állapotegyenletek következménye az alábbi egyenletrendszer:

$$(s+1)\cdot q_{s+1}=A\cdot q_s\cdot\left(1-\frac{r_s}{q_s}\right). \qquad (4)$$

$s=0, 1, \ldots, n-1$, ahol q_s annak a valószínűsége, hogy s db kiszolgáló foglalt, r_s pedig annak a valószínűsége, hogy s db kiszolgáló foglalt, és abból legalább k a mutató előtt.

Az r_s/q_s hányadosokat veszteségi tényezőknek is nevezik (Lásd pl. [2]; 82. o.) és a következő feltételes valószínűséggel egyeznek meg:

P (a következő igény elvész|s db kiszolgáló foglalt). Emiatt

$$s<k \quad \text{esetén} \quad r_s=0 \quad \text{és}$$

$$s\leq k \quad \text{esetén} \quad q_s=\frac{A^s}{s!}\cdot q_0. \qquad (5)$$

Ezek után rátérünk a két speciális eset ismertetésére.

A) $k=1$

Most lényeges lesz a kiszolgálók számának feltüntetése, ezért a korábbi q_s-t, r_s-t, x_i-t rendre $Q_s^{(n)}$, $R_s^{(n)}$, ill. $X_i^{(n)}$ fogja jelölni.

Tekintsük az $(n+1)$ kiszolgáló esetét. Mivel új igény érkezésekor a mutató eggyel tovább lép és a következő kiszolgálót lefoglalja, ha szabad volt, illetve nem változtatja meg annak foglalt voltát, ez azt jelenti, hogy a mutató előtti kiszolgáló állapotától bizonyos értelemben el lehet tekinteni.

Bevezetve ezért az $L^{(n+1)}=(l_{u+1}, \ldots, l_{u+n}, l_u)$ vektor helyett az $L_x^{(n+1)}=(l_{u+2}, \ldots, l_{u+n}, l_u)$ vektort, az erre vonatkozó állapot-átmenetek ugyanazok, mint az ugyanilyen koordinátákkal jellemzett n — kiszolgálós esetben — ugyanis kiszolgálás megszűnésekor a megfelelő helyen egy 1-esből 0 lesz, új igény érkezésekor pedig a megfelelő koordináta transzformáció:

$$\begin{array}{c}(\ ,\ ,\ ,\ \ldots,\ ,\ ,\ ) \\ \swarrow\ \swarrow\ \swarrow\ \quad \swarrow\ \swarrow \\ (\ ,\ ,\ ,\ \ldots,\ ,\ ,1)\end{array}$$

ciklikus eltolása 1-gyel a koordinátáknak, de az utolsó koordináta 1-es lesz.

Márpedig ez utóbbi transzformáció is független a kiszolgáló egységek számától. Emiatt a $p(L_x^{(n+1)})$-ekre ugyanazokat az egyenleteket lehet felírni, mint a megfelelő $p(L^{(n)})$-ekre, és az állapotegyenletek csak e valószínűségek arányát határozzák meg, ezért fennáll:

$$X_i^{(n+1)}+X_{i+2^n}^{(n+1)}=c\cdot X_i^{(n)}. \qquad (6)$$

Mégpedig azért ilyen alakban, mert a $p(L_x^{(n+1)})$ az két állapot valószínűségének összege; amikor $l_{u+1}=0$, illetve 1, és az i sorszám az előbbi állapothoz tartozik ($i<2^n$).

A c konstans úgy kapható meg, hogy a (6)-os egyenletet összegezzük i minden lehetséges értékére, és felhasználjuk (1')-t.

Ezért

$$(n+1)\cdot\left(X_i^{(n+1)}+X_{i+2^n}^{(n+1)}\right)=n\cdot X_i^{(n)}. \qquad (6')$$

Vegyük észre, hogy tetszőleges j esetén $X_j^{(n)}$ olyan állapotnak a valószínűségét jelöli, melyben a foglalt kiszolgálók száma egyenlő a j szám kettes számrendszerbeli alakjában a jegyek összegével, amit jelöljön $S(j)$. Ekkor pedig

$$\sum_{S(j)=s} n\cdot X_j^{(n)}=Q_s^{(n)} \quad \text{és hasonlóan}$$

$$\sum_{S(j)=s} (n+1)\cdot X_{j+2^n}^{(n+1)}=R_{s+1}^{(n+1)}.$$

Ezeket figyelembe véve, (6')-t olyan i-kre összegezve, melyekre $S(i)=s$, az következik, hogy

$$Q_s^{(n)}=Q_s^{(n+1)}-R_s^{(n+1)}+R_{s+1}^{(n+1)}, \tag{7}$$

(4)-ből kifejezhető $R_s^{(n)}$ a $Q_s^{(n)}$ és $Q_{s+1}^{(n)}$ segítségével és azt felhasználva (7)-ből a $\overline{Q}_j^{(n)}=Q_{n-j}^{(n)}$ jelöléssel az alábbit nyerjük:

$$\overline{Q}_j^{(n+1)}=\frac{A\cdot\overline{Q}_j^{(n)}+(n+2-j)\cdot\overline{Q}_{j-1}^{(n+1)}}{A+\mathrm{n}+1-j}. \tag{8}$$

Ez a rekurzió a $\overline{Q}_0^{(0)}=1$ és $\overline{Q}_j^{(n)}=0$ ha $j\notin[0,\mathrm{n}]$ feltételek figyelembevételével megoldható, és nyerjük a már ismertetett eredményt.

Az átvitt forgalom és a veszteség közötti $B=A\cdot(1-V)$ összefüggésből a foglaltsági eloszlás ismeretében már meghatározható a veszteség is, de a következő gondolatmenettel egyszerűbben kapjuk [4].

Mivel a mutató mindig eggyel lép tovább, annak valószínűsége, hogy egy hívás elvész, az ugyanaz, minthogy az őt n-nel megelőző hívás kiszolgálása még mindig tart (kihasználva az exponenciális eloszlás emlékezet nélküli tulajdonságát).

Ez pedig ugyanaz, mint

$$P(\xi_1+\ \dots\ +\xi_n<\eta),\qquad \text{ahol}\qquad \xi_i\sim\mathrm{Exp}(\lambda),$$

$\eta\sim\mathrm{Exp}(\mu)$ és ezek a valószínűségi változók függetlenek. Ez a valószínűség pedig

$$=\int_0^\infty\frac{\lambda^n\cdot x^{n-1}}{(n-1)!}\cdot e^{-(\lambda+\mu)x}\,\mathrm{d}x=\left(\frac{\lambda}{\lambda+\mu}\right)^n=\left(\frac{A}{A+1}\right)^n.$$

B) $k=\mathrm{n}-1$

Az (5) felhasználásával csak egyetlen veszteségi tényezőt kell meghatározni. Viszont most a (3)-as összefüggés nem más, mint $(A+\mathrm{n}-1)\cdot\mathrm{r}_{n-1}=q_n$, amiből (4) alapján meghatározható a q_n és q_{n-1} közti arány. $\sum_{s=0}^{n}q_s=1$-ből pedig az egész foglaltsági eloszlás meghatározható. A veszteség pedig

$$V_{n-1}(n)=\mathrm{r}_{n-1}+q_n.$$

IRODALOM

[1] L. Kleinrock: Qucueing Systems I., J. Wiicy and Sons (New-York), 1975.

[2] A Course in Teletraffic Engineering, Telecom Australia (Melbourne), 1978.

[3] Laborczi Zoltán: SIMULA 67 jegyzet, SZÁMKI Közlemények 13. (Budapest), 1976.

[4] Rét Andrásné: szóbeli közlése.