

Debreceni Egyetem

Informatika Kar

# Az M/G/1 sorbanállási rendszer és alkalmazásai

Témavezető:

Dr. Sztrik János

Professzor

Készítette

Kása Szabolcs

Programtervező Matematikus

Debrecen

2009

# Tartalomjegyzék

2.0 Az M/G/1 rendszer ismertetése.....	5
2.1 Az M/G/1 és M/M/1 rendszer legfőbb különbsége.....	5
2.2 Pollaczek-Khinchin általános formula.....	5
2.3 Példa az M/D/1 rendszeren keresztül.....	8
2.4 A sorhossz eloszlása M/G/1 rendszernél.....	9
2.4.1 Beágyazott Markov-láncok.....	9
2.4.2 Algoritmikus megközelítése a sorhossz eloszlásának.....	13
2.5 Összefoglaló az M/G/1 sorbanállási rendszerek képleteiről.....	16
3.0 Az M/G/1 rendszerekre, konkrét eloszlásokkal számoló analitikus program.....	17
3.1 A program működése.....	17
3.2 A program jellemzői.....	19
4.0 M/G/1 rendszer visszatérő inhomogén vakációkkal vizsgálata az IEEE 802.16e energia megtakarításos módszerhez.....	28
4.1 A Modell felépítése.....	28
4.2 A rendszerben lévő vakációk száma, a tétlen periódus és a foglalt periódus .....	29
5.0 Hosszú farkú osztott processzoros (PS) rendszerek.....	31
5.1 M/G/1/PS rendszer .....	31
5.2 Hosszú farkú osztott processzoros (PS) rendszerek türelmetlenséggel és áteresztés irányítás Packet Network hálózatokban.....	32
5.2.1 Az áteresztés irányítás alapértékei.....	33
5.2.2 Az RSR megközelítés.....	34
5.2.3 Szimulációs eredmények.....	35
5.2.4 M/G/s PS sor esetén.....	38
6.0 Konklúzió.....	39
7.0 Köszönetnyilvánítás.....	40
9.0 Irodalomjegyzék.....	41
10.0 Függelék:.....	42

## 1.0 Bevezetés

A dolgozat a középfokú sorbanállási rendszerekkel való számításokról ad egy kis betekintőt. A sorbanállási elmélet egy a napjainkban igen gyakran előforduló kellemetlen probléma.

Megjelenik a hétköznapi életben, mikor például egy boltban szeretnénk kiszámolni, hogy mikorra kerülünk sorra valamelyik pénztárnál, vagy egy bankban meddig kell várakoznunk, mire az ügyintéző velünk foglalkozik, de jelen van az informatikában is például ha egyszerre több programot futtatunk a számítógépen, vagy ha több hálózati csomag érkezik egy kiszolgálóhoz. Ha ismerjük a kiszolgálási időket és kiszolgálási szabályokat, akkor csökkenteni tudunk a várakozási időnkön, például ha van ismerősünk a sorban, gyorsabb processzort veszünk vagy magasabb sávszélességű hálózatot használunk.

A sorbanállás elmélet fontos szerepet játszik például a forgalomelméletben is, ami a távközlési hálózatok teljesítményanalízisének és tervezésének alaptudománya. Agner Kraup Erlang dán matematikus (1878- 1929) fogalmazta meg először a matematikai problémát. Az elmélet a telefonhálózatok fejlesztésével párhuzamosan fejlődött és lényegi elemévé vált a klasszikus távközlési hálózatok tervezésének.

A telefonhálózatok statikus jellege tette lehetővé, hogy olyan univerzális törvényt találjunk, mint a hívások keletkezésének Poisson természete. Ez a szabály kimondja, hogy a hívások függetlenek és a hívások közötti idő exponenciális eloszlású.

A sorbanállás megjelenik az egészségügyben is, A betegek számára az egészségügyi ellátás egyik igen fontos tényezője a szolgáltatás nyújtásához kapcsolódó várakozás. A várakozással töltött, elvesző idő mértéke a betegnek az ellátással kapcsolatos elégedettségét jelentősen befolyásolja, és kedvezőtlen esetben az ellátóról negatív kép alakulhat ki. A kiszolgálórendszer nem megfelelő méretezése ugyan akkor a szolgáltató számára felesleges többletköltséget generálhat, amely tovább rontja az intézmény pénzügyi helyzetét. Ahelyett, hogy egy-egy konkrét eset mennyiségi jellemzőit próbálnák megadni, a szolgáltatás egyensúlyi állapotát vizsgálják. Az igénybevevők sztochasztikus folyamat szerint érkeznek a kiszolgálási pontokra, ahol egy valószínűségi változó által meghatározott ideig tartó szolgáltatást kapnak. A két sztochasztikus

folyamat eredményeképpen alakul ki a várakozási sor, amelynek átbozsátási ideje gyorsabb kiszolgálást nyújtó rendszerrel csökkenthető.

Mint látjuk, a sorbanállás elméletet a hétköznapi életben tág környezetben jelenik meg.

Különbözső területeken nyerhetünk hasznos információkat a sorbanállás vizsgálatával.

A dolgozat az M/G/1 típusú rendszerekkel foglalkozik. Sorbanállási rendszerek elnevezéséhez

Kendall jelölésrendszerét használjuk. A jelölésrendszer alapján:

A rendszer leírható A/B/m/K/N ötössel, ahol

A: a beérkezési időközök eloszlásfüggvénye,

B: a kiszolgálási idő eloszlásfüggvénye,

m: a kiszolgálók száma,

K: a rendszer befogadóképessége, azaz a kiszolgálóegységben és a várakozási sorban tartózkodó igények maximális száma,

N: az igényforrás számossága.

A dolgozathoz mellékelek egy analitikus programot, amely a legjellemzőbb eloszlások mellett segít M/G/1 rendszerben lévő fontos adatok kiszámításához.

## 2. A Középfokú sorbanállási elmélet

### 2.1 Az M/G/1 rendszer ismertetése

Az egyetemi anyagból ismert **M/M/1** rendszerben mind a beérkezési, mind a kiszolgálási folyamat Markov típusú (exponenciális eloszlású). Az **M/G/1** rendszerénél viszont a kiszolgálás általános eloszlású, következésképpen új problémákba ütközünk, és ezekre más megoldási módszereket kell találnunk. Az **M/G/1** rendszerben 1 kiszolgálóegység van és a beérkezési folyamat  $\lambda$  paraméterű Poisson-folyamat, azaz a beérkezési időközök  $\lambda$  paraméterű exponenciális eloszlású valószínűségi változók. A kiszolgálási idő eloszlása tetszőleges lehet. A  $t$  időpontban a rendszerben tartózkodó igények száma -  $N(t)$  – önmagában nem alkot Markov-folyamatot. Az időegység alatt az  $N = n$  állapotból az  $N = n-1$  állapotba kerülés valószínűsége függ az igények a rendszerben eddig eltöltött idejétől is. Ezt az információt nem tartalmazza az  $N(t)$  érték. Ennek ellenére az átlagos sorhossz, a várakozási idő és a tartózkodási idő meghatározható. Ezen értékeket (a Pollaczek-Khinchin formulán keresztül) az alábbi pontokban származtathatjuk. A számítás alapja, hogy egy beágyazott Markov-láncot reprezentál a formula.

### 2.2 Pollaczek-Khinchin általános formula

A formula Pollaczek Félix Leo (Bécs, 1892. december 1. – Párizs Boulogne-Billancourt, 1981. április 29.) osztrák származású mérnök és matematikus, valamint Hincsin, Alekszandr Jakovlevics (Khinchin, Aleksandr Yakovlevich) (1894-1959) orosz matematikus nevéhez köthetjük. Kezdjük a számításokat a  $W$  várható várakozási idő megállapításával.  $W$  az igények a kiszolgálásra történő várakozásának ideje.

$$E[W] = E[Nq] * E[S] + E[R]$$

$R$  a maradék kiszolgálási idő.

Ahol:

$E[Nq]$  a várakozó igények száma

$E[S]$  átlagos kiszolgálási idő

Ez a két adat adja meg az igények átlagos kiszolgálási idejét.

$E[R]$  a befejezetlen munka a rendszerben – a kiszolgálás befejezéséhez szükséges idő

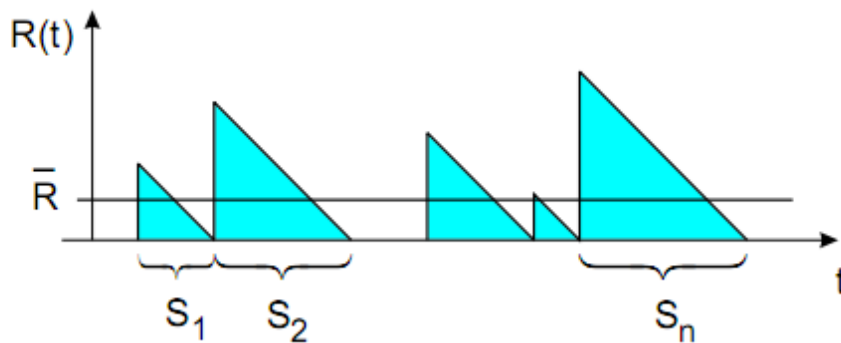
Ideális rendszerben az  $R = 0$ .

A Little - formula(*függelék*) alapján az átlagos sorhossz  $E[N_q]$  kiszámítható a várakozási idő segítségével. Az alábbi képlet alapján:

$$E[N_q] = \lambda E[W] \Rightarrow E[W] = \frac{E[R]}{1 - \rho}$$

$$\rho = \lambda * E[S]$$

A maradék kiszolgálási idő grafikus argumentumokkal könnyen következtethető. A grafikon reprezentálja a befejezetlen munkát az idő  $R(t)$  függvényében.



Tekintsünk  $t$ -t egy nagyobb méretű időintervallumnak. Az átlagértéket megkapjuk ha elosztjuk a háromszögek területeinek összegét az intervallum hosszával.

A háromszögek el vannak választva egymástól a tétlen időközökkel.

A háromszögek száma –  $n$  – meghatározható a  $\lambda$  beérkezési értékkel. Az átlagérték  $\lambda t$ .

Ezek alapján:

$$E[R] = \frac{1}{t} \int_0^t R(t') dt' = \frac{1}{t} \sum_{i=1}^n \frac{1}{2} S_i^2 = \frac{n}{t} \cdot \frac{1}{n} \cdot \sum_{i=1}^n \frac{1}{2} S_i^2$$

A Pollaczek-Khinchin általános képlet a várakozási időre:

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)}$$

Az átlagos várakozási időből könnyen megkaphatjuk az átlagos tartózkodási időt is.

$E[T] = E[S] + E[W]$ , ahol  $E[S]$  az igények saját kiszolgálási ideje.

Vezessük be a  $C_v^2$  értéket.

$$C_v^2 = V[S] / E[S]^2$$

$$E[S^2] = V[S] + E[S]^2 = (1 + C_v^2) \cdot E[S]^2$$

Ezek alapján a várakozási és tartózkodási idők képletei:

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)} = \frac{1 + C_v^2}{2} \cdot \frac{\rho}{1-\rho} \cdot E[S]$$

$$E[T] = E[S] + \frac{\lambda E[S^2]}{2(1-\rho)} = \left( 1 + \frac{1 + C_v^2}{2} \cdot \frac{\rho}{1-\rho} \right) \cdot E[S]$$

Valamint a várakozó igények és a rendszerben lévő igények száma:

$$E[N_q] = \lambda E[W] = \frac{\lambda^2 E[S^2]}{2(1-\rho)} = \frac{1 + C_v^2}{2} \cdot \frac{\rho^2}{1-\rho}$$

$$E[N] = \lambda E[T] = \lambda E[S] + \frac{\lambda^2 E[S^2]}{2(1-\rho)} = \rho + \frac{1 + C_v^2}{2} \cdot \rho + \frac{\rho^2}{1-\rho}$$

Megjegyzések a PK általános formulához:

- Az átlagérték függ a várakozástól  $E[S]$  és az rendszeridő eloszlás változótól  $V[S]$ , de nem a nagy értékeknél.
- Átlagértékek lineárisan nőnek a változóval
- A rendszertelenség magasabb várakozási időt és sorhosszt eredményez
- A formula hasonlít az  $M/M/1$ -éhez, eltérés a plusz  $(1+C_v^2)/2$  tényező

## 2.3 Példa az M/D/1 rendszeren keresztül

Az M/D/1 rendszer esetében, konstans kiszolgálási időnél  $V[S]=0 \Rightarrow C_v^2=0$  Ez alapján:

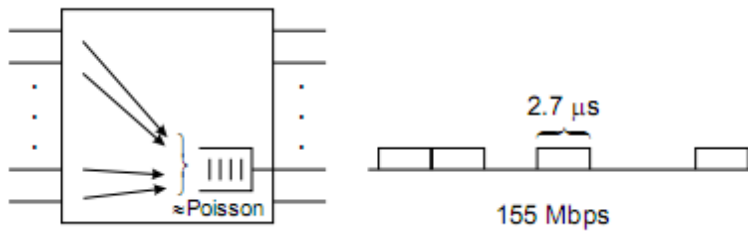
$$E[N] = \rho + \frac{1}{2} \frac{\rho^2}{1-\rho}$$

$$E[T] = \left(1 + \frac{1}{2} \frac{\rho}{1-\rho}\right) \cdot E[S]$$

Itt az eltérés az M/M/1-hez képest csak plusz  $\frac{1}{2}$  tényező.

Példaként modellezzünk egy ATM multiplexer kimeneti pufferét M/D/1 rendszerben. A konstans kiszolgálási idő most azt jelenti, hogy az ATM cellának fix mérete van (53 oktett) és az átviteli ideje szintén konstans. A kapcsolati sebessége 155 Mbit/s, ekkor az átviteli ideje  $S = 53 \cdot 8 / 155 \mu\text{s}$ .

Ábrán keresztül szemléltetve:



A kérdés, hogy mi a pufferben a cellák átlagértéke (beleértve a kézbesített cellákat is), valamint a cella átlagos tartózkodási ideje a pufferben, mikor a kapcsolat átlagos információs értéke 124 Mbit/s?

A kapcsolat kihasználtsága  $\rho = 124/155 = 0,8$ .

Ekkor:

$$E[N] = 0,8 + \frac{1}{2} \cdot \frac{0,8^2}{1-0,8} = 2,4$$

$$E[T] = \left(1 + \frac{1}{2} \cdot \frac{0,8^2}{1-0,8}\right) 2,7 \mu\text{s}$$

## 2.4 A sorhossz eloszlása M/G/1 rendszerénél

A sorhossz  $N(t)$  nem alkot Markov-láncot az M/G/1 rendszerben.

A rendszerben lévő igények száma nem adja meg, hogy milyen eloszlással érkeznek az igények. Viszont ez az eloszlás függ a már fogadott igények számától.

Ahogy fentebb is láttuk, az átlag sorhosszt könnyű kiszámítani. A sorhossz eloszlása is kiszámítható. Két különböző megközelítés van:

Az első annak a vizsgálatán alapszik, hogy a befejezetlen munka  $X_t$ , nem alkot Markov folyamatot. A Markov tulajdonság, nem egy beleső tulajdonsága a rendszernek.

Az  $X_t$  értéke a következőképpen írható le:

Mikor nincs beérkező igény, akkor  $X_t$  csökken egy  $C$  konstans értékre. (Ha  $X_t > 0$ ) Továbbá van egy valószínűség/időegység konstans,  $\lambda$ , az új érkezőnek, így a sor összmunkáját megkapjuk egy adott eloszlás mellett. Így nincs szükségünk az  $X_t$  eddigi értékére. A technikai nehézség ezzel az, hogy az  $X_t$  egy folyamatosan változó állapotú folyamat.

A második megközelítés szerint van egy beágyazott Markov-lánc, amivel az eloszlás kiszámítható. A következőkben ezt használjuk.

### 2.4.1 Beágyazott Markov-láncok

A beágyazott Markov-láncok használata M/G/1 rendszereknél az kiszolgálócsatornából való távozások időpontjaitól függ. Nem minden időpontot tudunk figyelni, csak bizonyos kiválasztott időpontokat. Ezeknek olyanoknak kell lenniük, hogy ha meghatározzuk az egyik ilyen időpontban a rendszerbeli igények számát, és figyelembe vesszük az elkövetkezendő beérkezéseket, akkor a legközelebbi alkalmas időpontban újra ki kell tudnunk számolni a rendszerbeli igények számának eloszlását, azaz valahogy le kell írni a kiszolgálócsatornában tartózkodó igény már eltelt kiszolgálási idejét. Sok ilyen ponttal van, de legcélszerűbb a kiszolgálócsatornából való távozási pillanatokból állókat tekintenünk. Ha megadjuk egy igény

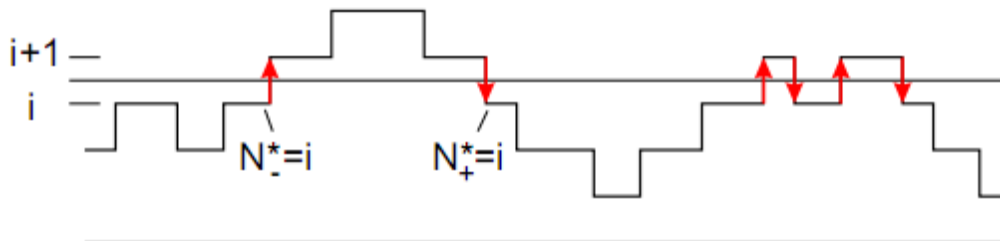
távozásakor a rendszerben maradt igények számát, akkor bármely jövőbeli pontban ismét ki tudjuk számolni ezt, hiszen a további beérkezések adottak. Ezekben a pillanatokban nulla a kiszolgálócsatornában tartózkodó igényre vonatkozó, már eltelt kiszolgálási idő, hiszen ez az igény (ha van ilyen) éppen az adott pillanatban lépett be a csatornába. Leírásunk nem más, mint egy szemi-Markov folyamat. A szemi-Markov folyamat esetében az egy helyben maradás ideje tetszőleges eloszlású lehet, míg Markov folyamatnál exponenciális eloszlást követelünk meg. Ezekben a pillanatokban egy beágyazott Markov-láncot definiálunk, ami a rendszerben tartózkodó igények száma közvetlenül a távozás után. Állapotváltozások csak a beágyazott pontokban jöhetnek létre, és diszkrét állapotteret alkotnak.

A rendszer állapotát a rendszerben tartózkodó igények számaként értelmezve megfigyelhetjük a rendszer állapotváltozásait az idő függvényében. Ezek a változások közvetlen szomszéd típusúak. Ha a rendszer elég sokáig működik, akkor a felfelé irányuló átlépések relatív gyakoriságának meg kell egyeznie a lefelé irányuló átmenetek relatív gyakoriságával.

Legyen  $N$  a sor hossza a tetszőleges időpillanatban,

$N^*$  a sor hossza az igény érkezése után

és  $N^+$  a sor hossza egy kiszolgált igény után.



Tudjuk még, hogy Poisson-folyamatok esetén  $N \sim N^*$ . (Mivel: függelék PASTA összefüggés)

Általában minden rendszernél az érkezés előtti állapot és a távozás utáni állapot közelítőleg megegyezik.  $N^* \sim N^+$ .

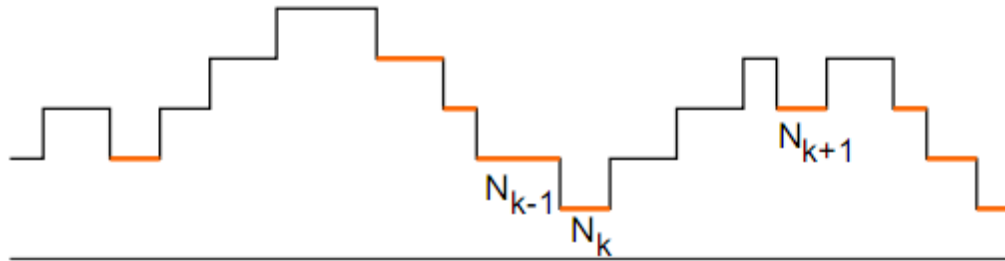
A két megállapítás alapján  $N \sim N^+$ .

Tehát beláttuk, hogy miért fontos egy adott pillanatban az  $N$  eloszlásához a távozó igények

eloszlása. Innentől a  $N^*_+$  Markov-lánccal foglalkozunk, amit az egyszerűség kedvéért jelöljünk csak  $N$ -nel.

Nevezzük még el  $N_k$ -t a sor hosszának a  $k$ . igény kiszolgálása után.

Valamint  $V_k$ -t a beérkezett igények számának, amíg a  $k$ . igényt szolgálja ki a rendszer.



Állítás: A diszkrét idejű  $N_k$  folyamat egy Markov-lánccot képez. (ha az nem egy születési-kihalási folyamat)

Bizonyítás: Adott  $N_k$ -t és  $N_{k+1}$ -t határozottan ki lehet fejezni egy random  $V_{k+1}$  változó segítségével, ami független  $N_k$ -tól.

$$N_{k+1} = N_k - 1 + V_{k+1}, \text{ ha } N_k \geq 1 \text{ és } N_{k+1} = V_{k+1}, \text{ ha } N_k = 0$$

Ha  $N_k > 0$ , akkor  $k$  kiszolgálása után,  $k+1$ . igény már a sorban van és érkezik rögtön a kiszolgálóhoz. Végül mikor  $k+1$ . igény is kiszolgálásra kerül, akkor a sorhossz csökken eggyel. Közben (a  $k+1$ . igény kiszolgálása alatt),  $V_{k+1}$  érkező van.

Ha  $N_k = 0$ , a  $k$ . igény elhagyja az üres sort. Ha  $k+1$ . igény megérkezik akkor a sorhossz megnő 1-el és a  $k+1$ . igény kiszolgálása után csökken eggyel. A sor tartalmazza azokat az igényeket, melyek a  $k+1$ . igény kiszolgálása alatt érkeznek.

A kiszolgálási idők függetlenek és az érkezések Poisson eloszlásúak.

Legyen:

$$\check{N}_k = (N_k - 1)^+ = N_k - 1, \text{ ha } N_k \geq 1 \text{ vagy } N_k, \text{ ha } N_k = 0$$

Ekkor:

$$N_{k+1} = \check{N}_k + V_{k+1} + 1$$

Ha  $\mathbf{V}$  és  $\check{\mathbf{N}}$  függetlenek, határozzuk meg a generátor függvényeiket.

$$\mathbf{GN}(\mathbf{z}) = \mathbf{G}\check{\mathbf{N}}(\mathbf{z}) \cdot \mathbf{GV}(\mathbf{z})$$

Vegyük  $\mathbf{G}\check{\mathbf{N}}(\mathbf{z})$ -t és  $\mathbf{GV}(\mathbf{z})$ -t.

$$G_{\check{N}}(Z) = \frac{G_N(Z) - (1-\rho)(1-z)}{z}$$

$$G_{\check{N}}(z) = \frac{G_N(z) - 1(1-\rho)(1-z)}{z}$$

Ahol  $\rho = \lambda E[S]$ .

Legyen  $\mathbf{X}$  egy random változó. Meg kell határoznunk a beérkező igények eloszlását,  $\mathbf{K}$ -t. Ha  $\mathbf{K}$   $\lambda$  paraméterű Poisson-folyamat  $\mathbf{X}$  intervallumon és a generátor függvénye  $\mathbf{GK}(\mathbf{z})$ .  $\mathbf{S}^*(\mathbf{0})$  pedig a kiszolgálási idő Laplace transzformáltja:

$$G_K(z) = E[z^K] = E[z^K \vee X] = E[e^{-(1-z)\lambda X}] = X * ((1-z)\lambda), X^*(s) = E[e^{-sX}]$$

Általános esetben:  $G_V(z) = S * ((1-z)\lambda)$

Speciális esetben:  $G_K(z) = X * ((1-z)\lambda)$

$\mathbf{N}$  generátorfüggvénye ezek alapján:

$$G_N(z) = \frac{(1-\rho)(1-z)}{S * ((1-z)\lambda) - z} \cdot S * ((1-z)\lambda) = \frac{(1-\rho)(1-z)}{1-z/S * ((1-z)\lambda)}$$

Pollaczek-Khichin formulán keresztül a tartózkodási idő eloszlása:

$$T^*(s) = (1-\rho) \frac{s}{s - \lambda + \lambda S^*(s)} S^*(s)$$

Ahol  $\mathbf{T}^*(\mathbf{0})$  a tartózkodási idő Laplace transzformáltja.

Veszünk egy szabad változót  $\mathbf{z}$ -t. Legyen  $\mathbf{z} = \mathbf{1} - \mathbf{s}/\lambda$ , átalakítva  $\mathbf{s} = (\mathbf{1} - \mathbf{z})\lambda$ .

Pollaczek-Khinchin formulán keresztül a várakozási idő:

$$W^*(s) = \frac{(1-\rho)s}{s-\lambda+\lambda S^*(s)}$$

Ahol  $W^*(s)$  a várakozási idő Laplace transzformáltja és  $S^*(s)$  a kiszolgálási idő Laplace transzformáltja.

## 2.4.2 Algoritmikus megközelítése a sorhossz eloszlásának

Az előző fejezetben a PK formula segítségével a sorhossz eloszlásának generátorfüggvényét határoztuk meg. Ez az eredmény elméletben fontos. Eloszlások momentumainak kiszámításához használhatjuk őket. Azonban a formula nem teljesen praktikus, mert a z-től való függés miatt komplikált a számolás. A sorhossz eloszlás viszont algoritmikusan is számítható. Ebben a fejezetben az algoritmust származtatjuk.

Először vegyük át újra az  $N_k$  beágyazott Markov-láncot. Tudjuk, hogy:

$$N_{k+1} = N_k - 1 + V, \text{ ha } N_k \geq 1 \text{ és } N_{k+1} = V, \text{ ha } N_k = 0$$

ahol V az érkező igények száma a k+1. igény kiszolgálási ideje alatt.

Vezessük be az  $f_S(x)$  jelölést az S kiszolgálási idő sűrűségfüggvényére és legyen  $k_i = P(V=i)$ .

A teljes valószínűség törvénye alapján:

$$k_i = P(V=i) = \int_0^{\infty} P(V=i \vee S=x) f_S(x) dx = \int_0^{\infty} \frac{(\lambda x)^i}{i!} e^{-\lambda x} f_S(x) dx$$

így:

$$k_i = \int_0^{\infty} \frac{(\lambda x)^i}{i!} e^{-\lambda x} f_S(x) dx \quad i = 0, 1, \dots$$

Mikor a beérkezési intenzitás  $\lambda$  és az  $f_S(x)$  adott, akkor a  $k_i$  könnyen kiszámítható numerikusan is.

Nézzük meg a beágyazott lánc átmeneti valószínűségi mátrixát!

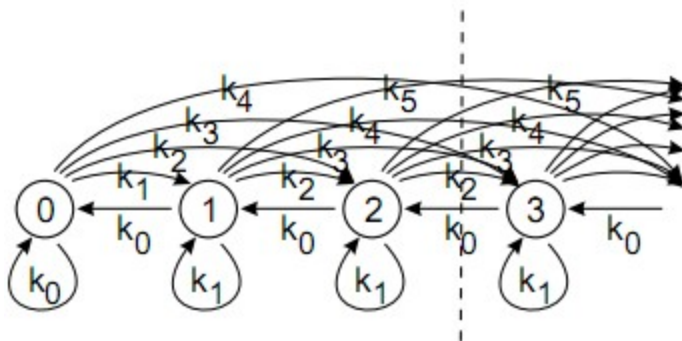
A mátrix értékei:

$$p_{ij} = P(N_{k+1} = j \vee N_k = i) = k_{j-i+1} \text{ ha } i \geq 1 \text{ vagy } p_{ij} = k_{j-i}, \text{ ha } i = 0$$

Az állapot átmeneti mátrix:

$$\mathbf{P} = \begin{pmatrix} k_0 & k_1 & k_2 & k_3 & \dots \\ k_0 & k_1 & k_2 & k_3 & \dots \\ 0 & k_0 & k_1 & k_2 & \dots \\ 0 & 0 & k_0 & k_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Az állapot átmeneti diagramm:



A rekurzív egyenlet metódusa:

$$\begin{cases} k_0 \pi_1 = (k_1 + k_2 + \dots) \pi_0 \\ k_0 \pi_2 = (k_2 + k_3 + \dots) \pi_0 + (k_2 + k_3 + \dots) \pi_1 \\ k_0 \pi_3 = (k_3 + k_4 + \dots) \pi_0 + (k_3 + k_4 + \dots) \pi_1 + (k_2 + k_3 + \dots) \pi_2 \\ \vdots \\ k_0 \pi_i = (k_i + k_{i+1} + \dots) \pi_0 + (k_i + k_{i+1} + \dots) \pi_1 + \dots + (k_2 + k_3 + \dots) \pi_{i-1} \end{cases}$$

Amivel a  $\pi_i$ -k könnyen kiszámolhatóak. Kezdőértékként  $\pi_0 = 1 - \rho$

Vezessük be a  $a_i = k_{i+1} + k_{i+2} + k_{i+3} + \dots = P(V > i)$  jelölést. Azaz  $a_i$  annak a valószínűsége, hogy S kiszolgálási idő alatt legalább  $i+1$  igény érkezik. Ennek függvényében az általános rekurziós

lépés megadható az alábbi formában:

$$\pi_i = \frac{1}{k_0} \left( a_{i-1} \pi_0 + \sum_{j=1}^{i-1} a_{i-j} \pi_j \right) \quad \text{Mivel } k_0 + k_1 + k_2 + \dots = 1,$$

$$\text{így } k_0 = 1 - (k_1 + k_2 + \dots) = 1 - a_0$$

A rekurzió indul  $\pi_0 = 1 - \rho$ -től.

Fontos egynek, hogy kiértékeljük az  $a_i$  sorozat összegeket, de tudunk származtatni egy integrálos kifejezést is. Jelöljük el  $X_j$ -vel egy  $\lambda$  intenzitású Poisson-folyamat általános beérkezési idejét.  $X_j \sim \exp(\lambda)$ . Ekkor:

$$a_i = P(S > X_1 + X_2 + X_3 + \dots + X_{i+1})$$

Az  $X_1 + X_2 + X_3 + \dots + X_{i+1}$  összeg Erlang( $i+1, \lambda$ ) eloszlású. Asűrűségfüggvénye  $e^{-\lambda x} (\lambda x)^i / i!$ .

Tudjuk még, hogy  $G_s(x) = 1 - F_s(x)$ . A teljes valószínűség tétele alapján:

$$a_i = \int_0^{\infty} G_s(x) \frac{(\lambda x)^i}{i!} \lambda e^{-\lambda x} dx = \int_0^{\infty} G_s(y/\lambda) \frac{y^i}{i!} e^{-y} dy$$

Ebből  $a_i$  numerikusan is kiszámítható.

## 2.5 Összefoglaló az M/G/1 sorbanállási rendszerek képleteiről

Az eddigi számítások segítségével meg tudjuk kapni az M/G/1 rendszer sorhosszának eloszlását. A legfontosabb M/G/1 rendszerekre a következő eredmények adódnak:

Sorbanállási rendszer	Sorhossz eloszlás $P(N=n)$
M/M/1	$(1 - \rho)\rho^n$
M/H <sub>2</sub> /1	$q(1 - \alpha_1)\alpha_1^n + (1 - q)(1 - \alpha_2)\alpha_2^n$
M/D/1	$(1 - \rho)\sum_{k=0}^n e^{k\rho} (-1)^{n-k} \frac{(k\rho + n - k)(k\rho)^{n-k-1}}{(n - k)!}$
M/E <sub>k</sub> /1	$(1 - \rho)\sum_{j=0}^n (-1)^{n-j} \frac{\alpha^{n-j-1}}{(1 - \alpha)^{kj}} \left[ \binom{kj}{n-j} \alpha + \binom{kj}{n-j-1} \right]$

A táblázatban H<sub>2</sub> a hiperexponenciális eloszlást jelent  $\alpha_1$  és  $\alpha_2$  és  $p$  paraméterekkel. E<sub>k</sub> jelenti a  $k$  paraméterű Erlang eloszlást  $\alpha$  és  $k$  paraméterekkel. Ez a két eloszlás azért fontos, mert segítségével tetszőleges M/G/1 rendszert közelíthetünk. Ahol a kiszolgálási idő relatív szórásnégyzete 1 vagy az alatt van az M/E<sub>k</sub>/1, ahol ez az érték 1 felett van, ott M/H<sub>2</sub>/1 rendszert használjuk az approximációhoz.

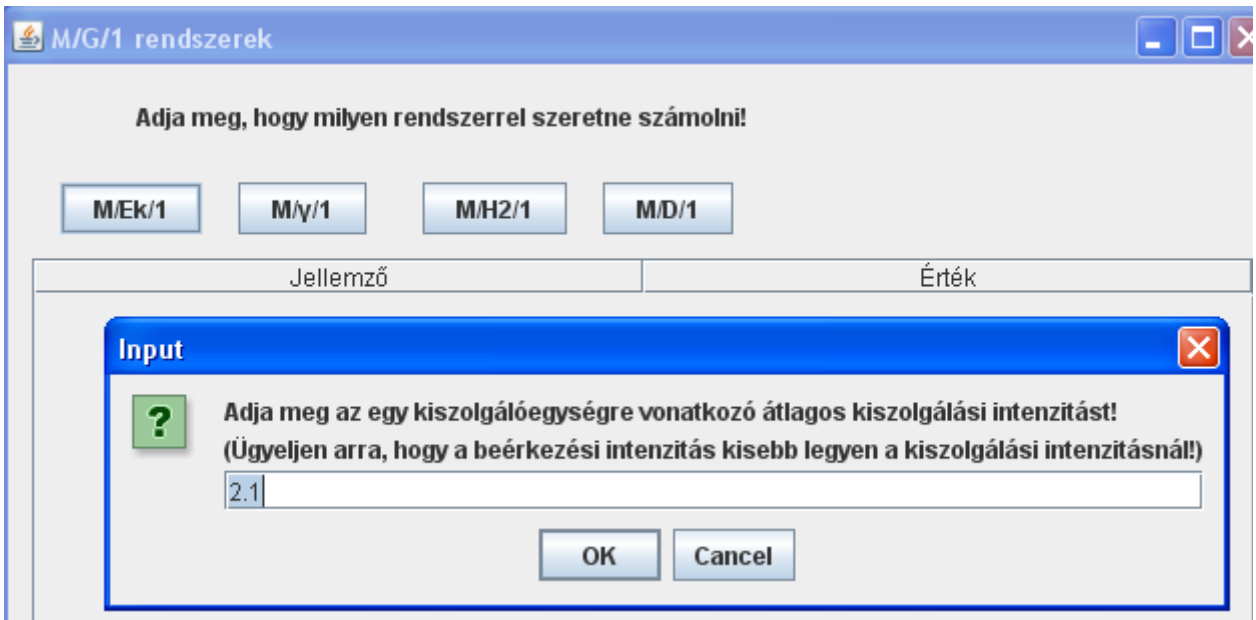
### 3.0 Az M/G/1 rendszerekre, konkrét eloszlásokkal számoló analitikus program.

#### 3.1 A program működése.

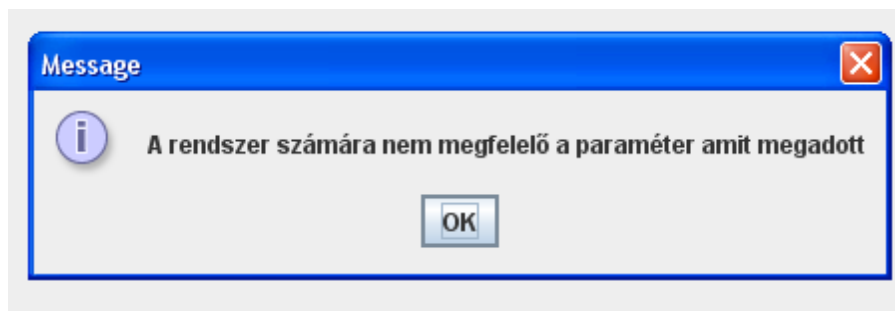
A dokumentumhoz tartozik egy program ami az M/D/1, M/E<sub>k</sub>/1, M/H<sub>2</sub>/1 és az M/γ/1 rendszerekkel való számításokban segít. Ezek a leginkább ismert M/G/1 rendszerek. A Programban először ki kell választanunk, hogy melyik rendszerben szeretnénk számolni:



A megfelelő gomb megnyomása után a program bekéri a rendszer paramétereit. Ezt előugró ablakok segítségével tehetjük meg.



Fontos, hogy figyeljünk arra, hogy a beérkezési intenzitás kisebb legyen, mint a kiszolgálási intenzitás! Ha ez a feltétel nem teljesül, vagy pedig nem számot adunk meg paraméternek, akkor az alábbi hibaüzenettel kilép a program:



Ha sikeresen megadtuk a paramétereket, akkor a program az alábbi formában adja meg az adott paraméterek melletti rendszerjellemzőket:

**M/G/1 rendszerek**

Adja meg, hogy milyen rendszerrel szeretne számolni!

Jellemző	Érték
M/Ek/1 rendszer	
Rendszerbe jövő igények beérkezési intenzitása	1.0
Az egy kiszolgálóegységre vonatkozó átlagos kiszolgálási ideje	2.1
Az Erlang eloszlás paramétere	2.0
Az igény átlagos kiszolgálási ideje	0.47619047619047616
A kiszolgáló egység kihasználtsága	0.47619047619047616
s n-ik momentuma	0.3401360544217687
s második momentuma	0.3401360544217687
s harmadik momentuma	0.32393909944930344
Egyensúlyi állapotban valamely igény átlagos várakozási ideje	0.3246753246753247
Egyensúlyi állapotban a sorbanálló igények átlagos várakozási ideje	0.3246753246753247
Nq szórásnégyzete	0.6362324544142726
Várható várakozási idő	0.6818181818181817
q második momentuma	0.41697119619197537
Sorbanállási idő szórásnégyzete	0.3115571297389479
Stacionárius esetben valamely igénynek a rendszerben töltött idő szórásnégyzete	0.8008658008658008
Stacionárius állapotban a rendszerben tartózkodó igények átlagos várakozási ideje	0.8008658008658008
N szórásnégyzete	1.2258016154120046
w második momentuma	1.0663218455426247
A rendszerben töltött idő szórásnégyzete	0.4249358145462041

Ha újra szeretnénk számolni a programmal, azt a gombok újbóli megnyomásával tehetjük meg.

### 3.2 A program jellemzői

A programot Java Swingben írtam, BetBeans IDE 6.7.1 toolkit segítségével.

Az egyes rendszerek képletei pedig a következők:

#### M/E<sub>k</sub>/1 rendszer:

Az igény átlagos kiszolgálási ideje:

Megegyezik az egy kiszolgálóegységre vonatkozó átlagos kiszolgálási intenzitás reciprokával.

(jele:  $W_s$ )

S n-ik momentuma:

$$E[s^n] = \left(1 + \frac{1}{k}\right) \left(1 + \frac{2}{k}\right) \dots \left(1 + \frac{n-1}{k}\right) W_s^n, n = 1, 2, \dots$$

S második momentuma:

$$E[s^2] = \left(1 + \frac{1}{k}\right) W_s^2$$

S harmadik momentuma:

$$E[s^3] = \left(1 + \frac{1}{k}\right) \left(1 + \frac{2}{k}\right) W_s^3$$

Egyensúlyi állapotban valamely igény átlagos várakozási ideje:

$$W_q = \lambda E \frac{[s^2]}{2(1-\rho)}$$

Egyensúlyi állapotban a sorbanálló igények átlagos száma:

$$L_q = \lambda \cdot W_q$$

$N_q$  szórásnégyzete:

$$\sigma_{N_q}^2 = \frac{\rho^2(1+k)}{2k(1-\rho)} \left[ 1 + \frac{\rho^2(1+k)}{2k(1-\rho)} + \frac{2\rho(k+2)}{3k} \right]$$

Várható várakozási idő:

$$E[q \vee q > 0] = \frac{W_s}{1-\rho} \left( 1 + \frac{1}{k} \right)$$

Q második momentuma:

$$E[q^2] = 2W_q^2 + \frac{\rho W_s^2(k+1)(k+2)}{3k^2(1-\rho)}$$

Sorbanállási idő szórásnégyzete:

$$\sigma_q^2 = E[q^2] - W_q^2$$

Stacionárius esetben valamely igénynek a rendszerben való tartózkodási ideje:

$$W = W_q + W_s$$

Stacionárius állapotban a rendszerben tartózkodó igények átlagos száma:

$$L = \lambda W$$

N szórásnégyzete:

$$\sigma_N^2 = \frac{\rho^3(k+1)(k+2)}{3k^2(1-\rho)} + \left( \frac{\rho^2 \left(1 + \frac{1}{k}\right)}{2(1-\rho)} \right)^2 + \frac{\rho^2(3-2\rho) \left(1 + \frac{1}{k}\right)}{2(1-\rho)} + \rho(1-\rho)$$

W második momentuma:

$$E[w^2] = E[q^2] + \frac{W_s^2 \left(1 + \frac{1}{k}\right)}{1-\rho}$$

A rendszerben töltött idő szórásnégyzete:

$$\sigma_w^2 = E[w^2] - W^2$$

**Az M/D/1 rendszer:**

Az igény átlagos kiszolgálási ideje:

Megegyezik az egy kiszolgálóegységre vonatkozó átlagos kiszolgálási intenzitás reciprokával.

(jele:  $W_s$ )

A kiszolgáló egység kihasználtsága:

$$\rho = \lambda W_s$$

S n-ik momentuma:

$$E[s^n] = W S^n, n=1,2,\dots$$

Annak a stacionárius valószínűsége, hogy a rendszer üres:

$$\rho_0 = 1 - \rho$$

Annak a stacionárius valószínűsége, hogy a rendszerben 1 igény van:

$$\rho_1 = (1 - \rho)(e^\rho - 1)$$

Annak a stacionárius valószínűsége, hogy a rendszerben n igény van:

$$\rho_n = (1 - \rho) \sum_{n=0}^n \frac{(-1)^{n-j} (j\rho)^n - j - 1 (j\rho + n - j) e^{j\rho}}{(n-j)!}, n = 2, 3 \dots$$

Egyensúlyi állapotban valamely igény átlagos várakozási ideje:

$$W_q = \frac{\rho W_s}{2(1 - \rho)}$$

Q második momentuma:

$$E[q^2] = 2W_q^2 + \frac{\rho W_s^2}{3(1 - \rho)}$$

Sorbanállási idő szórásnégyzete:

$$\sigma_q^2 = E[q^2] - W_q^2$$

Egyensúlyi állapotban a sorbanálló igények átlagos száma:

$$L_q = \lambda W_q$$

$N_q$  szórásnégyzete:

$$\sigma_{Nq}^2 = \frac{\rho^2}{3(1 - \rho)} + \left[ \frac{\rho^2}{2(1 - \rho)} \right]^2 + \frac{\rho^2}{2(1 - \rho)}$$

Stacionárius esetben valamely igénynek a rendszerben való tartózkodási ideje:

$$W = W_q + W_s$$

W második momentuma:

$$E[w^2] = E[q^2] + \frac{W_s^2}{1-\rho}$$

A rendszerben töltött idő szórásnégyzete:

$$\sigma_w^2 = E[w^2] - W^2$$

Stacionárius állapotban a rendszerben tartózkodó igények átlagos száma:

$$L = \lambda W$$

N szórásnégyzete:

$$\sigma_N^2 = \frac{\rho^3}{1-\rho} + \left( \frac{\rho^2}{2(1-\rho)} \right)^2 + \frac{\rho^2(3-2\rho)}{2(1-\rho)} + \rho(1-\rho)$$

**M/γ/1**

Az igény átlagos kiszolgálási ideje:

Megegyezik az egy kiszolgálóegységre vonatkozó átlagos kiszolgálási intenzitás reciprokával.

(jele:  $W_s$ )

A kiszolgáló egység kihasználtsága:

$$\rho = \lambda W_s$$

S n-ik momentuma:

$$E[s^n] = \beta(\beta+1) \cdot \dots \cdot (\beta+n-1), n=1,2,3 \dots$$

S második momentuma:

$$E[s^2] = W_s^2(1 + C_s^2), \text{ ahol } C_s^2 = 1/\beta$$

S harmadik momentuma:

$$E[s^3] = W_s^3 (1 + C_s^2) (1 + 2C_s^2)$$

Egyensúlyi állapotban valamely igény átlagos várakozási ideje:

$$W_q = \frac{\rho W_s}{2(1-\rho)}$$

Egyensúlyi állapotban a sorbanálló igények átlagos száma:

$$L_q = \lambda W_q$$

Nq szórásnégyzete:

$$\sigma_N^2 = \frac{\rho^2 (1 + C_s^2)}{2(1-\rho)} \left[ 1 + \frac{\rho^2 (1 + C_s^2)}{2(1-\rho)} + \frac{2\rho (1 + 2C_s^2)}{3} \right]$$

Várható várakozási idő:

$$E[q | q > 0] = \frac{W_s}{1-\rho} \left( 1 + \frac{C_s^2}{2} \right)$$

Q második momentuma:

$$E[q^2] = 2W_q^2 + \frac{\rho W_s^2 (1 + C_s^2) (1 + 2C_s^2)}{3(1-\rho)}$$

Sorbanállási idő szórásnégyzete:

$$\sigma_p^2 = E[q^2] - W_q^2$$

Stacionárius esetben valamely igénynek a rendszerben való tartózkodási ideje:

$$W = W_q + W_s$$

Stacionárius állapotban a rendszerben tartózkodó igények átlagos száma:

$$L = \lambda W$$

N szórásnégyzete:

$$\sigma_N^2 = \frac{\rho^3(1+C_S^2)(1+2C_S^2)}{3(1-\rho)} + \left( \frac{\rho^2(1+C_S^2)}{2(1-\rho)} \right)^2 + \frac{\rho^2(3-2\rho)(1+C_S^2)}{2(1-\rho)} + \rho(1-\rho)$$

W második momentuma:

$$E[w^2] = E[q^2] + \frac{W_s^2(1+C_S^2)}{1-\rho}$$

A rendszerben töltött idő szórásnégyzete:

$$\sigma_w^2 = E[w^2] - W^2$$

### M/H<sub>2</sub>/1

Legyenek  $z_1$  és  $z_2$  az alábbi egyenlet gyökei:

$$\rho_1 \rho_2 z^2 - (\rho_1 + \rho_2 + \rho_1 \rho_2)z + 1 + \rho_1 + \rho_2 - \rho = 0$$

$$C_1 = \frac{(z_1 - 1)(1 - \rho z_2)}{z_1 - z_2}$$

$$C_2 = \frac{(z_2 - 1)(1 - \rho z_1)}{z_2 - z_1}$$

A kiszolgáló egység kihasználtsága:

$$\rho = \lambda W_s$$

Annak a stacionárius valószínűsége, hogy a rendszerben n igény van:

$$\rho_n = C_1 z_1^{-n} + C_2 z_2^{-n} \quad n = 0, 1, \dots$$

Annak a stacionárius valószínűsége, hogy a rendszer üres:

$$\rho_0 = 1 - \rho$$

Annak a stacionárius valószínűsége, hogy a rendszerben n vagy több igény van:

$$P(N \geq n) = C_1 \frac{z_1^{-n+1}}{z_1 - 1} - C_2 \frac{z_2^{-n+1}}{z_2 - 1}$$

A q eloszlásfüggvénye:

$$W_q[t] = P(q \leq t) = 1 - C_5 e^{-at} - C_6 e^{-bt}, t \geq 0, \text{ ahol } a = -\zeta_1, b = -\zeta_2.$$

$\zeta_1$  és  $\zeta_2$  gyökei az alábbi egyenletnek:

$$\zeta^2 + (\mu_1 + \mu_2 - \lambda)\zeta + \mu_1 \mu_2 (1 - \rho) = 0$$

$$C_5 = \frac{\lambda(1 - \rho)\zeta_1 + \rho(1 - \rho)\mu_1 \mu_2}{a(\zeta_1 - \zeta_2)}$$

$$C_6 = \frac{\lambda(1 - \rho)\zeta_2 + \rho(1 - \rho)\mu_1 \mu_2}{a(\zeta_2 - \zeta_1)}$$

Egyensúlyi állapotban valamely igény átlagos várakozási ideje:

$$W_q = \frac{\lambda E[s^2]}{2(1 - \rho)}$$

Várható várakozási idő:

$$E[q | q > 0] = \frac{W_s}{1 - \rho} \left( \frac{1}{2} + \frac{C_s^2}{2} \right)$$

q második momentuma:

$$E[q^2] = 2W_q^2 + \frac{\lambda E[s^3]}{3(1 - \rho)}$$

$$\text{Ahol } E[s^3] = \frac{6q_1}{\mu_1^3} + \frac{6q_2}{\mu_2^3}$$

A sorbanállási idő szórásnégyzete:

$$\sigma_p^2 = E[q^2] - W_q^2$$

Stacionárius esetben valamely igénynek a rendszerben való tartózkodási ideje:

$$W = W_q + W_s$$

w második momentuma:

$$E[w^2] = E[q^2] + \frac{E[s^2]}{1-\rho}$$

s második momentuma:

$$E[s^2] = \frac{2q_1}{\mu_1^2} + \frac{2q_2}{\mu_2^2}$$

A rendszerben eltöltött idő szórásnégyzete:

$$\sigma_w^2 = E[w^2] - W^2$$

Egyensúlyi állapotban a sorban álló igények átlagos száma:

$$L_q = \lambda W$$

Nq szórásnégyzete

$$\sigma_{Nq}^2 = \frac{\lambda^3 E[s^3]}{3(1-\rho)} + \left(\frac{\lambda^2 E[s^2]}{2(1-\rho)}\right)^2 + \frac{\lambda^2 E[s^2]}{2(1-\rho)}$$

Stacionárius állapotban a rendszerben tartózkodó igények átlagos száma:

$$L = \lambda W$$

N szórásnégyzete

$$\sigma_N^2 = \frac{\lambda^3 E[s^3]}{3(1-\rho)} + \left(\frac{\lambda^2 E[s^2]}{2(1-\rho)}\right)^2 + \frac{\lambda^2(3-2\rho)E[s^2]}{2(1-\rho)} + \rho(1-\rho)$$

## **4.0 M/G/1 rendszer visszatérő inhomogén vakációkkal vizsgálata az IEEE 802.16e energia megtakarításos módszerhez.**

### **4.1 A Modell felépítése**

Ezen fejezet egy általános közelítést a visszatérő inhomogén vakációkkal rendelkező sorbanállási modellek közelítéséhez. A vakáció végeztével a kiszolgáló egy másik vakációra tér át, ami lehet más valószínűségű eloszlás is ha az előző vakáció alatt nem érkezett igény. Abban az esetben ha érkezett legalább egy igény a vakáció alatt, akkor egy rövid felkészülési idő múlva a rendszer foglalt állapotba kerül. Hogy jól nyomon tudjuk követni a rendszer paramétereit, vizsgáljunk egy egyszerű M/G/1 rendszert. Az elméleti modellt alkalmazzák energia megtakarítás céljából mobil egységeknél, amiknél a kikapcsolt üzemmód időtartama megfelel egy vakációnak. Különböző rendszerek teljesítmény mérőszámai, mint például a frame válaszsideje és az energia elosztása kiszámíthatóak. Egy korlátozott optimalizálási probléma feladat maximalizálni az energiával való gazdálkodást energia takarékos módban. A tanulmányunknak köszönhetően meg tudjuk majd adni azokat a paramétereket, melyekkel a rendszer legrosszabb esetben is a lehető legjobb teljesítményt nyújtja.

Az energiatakarékos mód a kulcsa a megfelelő energiakihasználásnak a mobil technológiáknál, mint például Wifi, 3G és WiMaX. Az energiatakarékos mód növeli az eszközök élettartamát is. Az IEEE 802.16e különböző típusú energiatakarékos rendszereket is magába foglal.

A továbbiakban vizsgáljuk meg a vezeték nélküli csatornát és a mobil csomópontokat mintázó M/G/1 rendszert. Az M/G/1 rendszerünk egy meghatározott ideig vakáció állapotba kerül amikor a sor üres. Amint vége a vakáció állapotnak, egy új vakáció indul el addig, míg nem érkezik igény a rendszerbe. Az igények  $\lambda$  paraméterű Poisson-folyamat szerint érkeznek. Nevezzük el a folyamatot:  $N(t)$ ,  $t \geq 0$ -nak. Legyen  $\sigma$  egy ugyanolyan eloszlású random változó, mint sor kiszolgálási ideje. Figyeljük meg, hogy a sor hossza egy foglalt periódus kezdeténél hatással van a foglalt állapot időtartamára. Mivel a folyamat Poisson, ezért a sor ciklikusan regenerálódik mindig mikor üres lesz. A regenerálódási ciklus áll egy tétlen periódusból, legyen  $I$  egy random

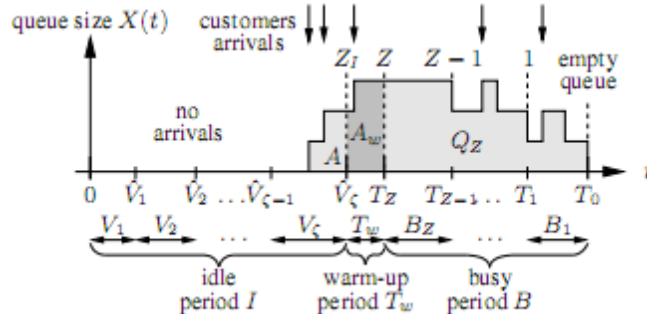
változó olyan eloszlással, mint a tétlen periódusok. Az általános I tétlen periódus  $\zeta$  vakáció periódusokból áll. Jelöljük el ezeket:  $V_1, \dots, V_\zeta$ . A ciklus második periódusa a felkészülési periódus. Ennek fix  $T_w$  időtartama van, míg a kiszolgáló felkészül kiszolgálni az igényeket. A ciklus harmadik periódusa a foglalt periódus. Legyen  $B$  egy random változó, olyan eloszlással mint amivel a sor rendelkezik foglalt állapotban.  $V_i$  eloszlása  $i$ -től függ, tehát a vakációk nem biztos hogy egyforma eloszlásúak. Függetlenek egymástól. Legyen  $X(t)$  a  $t$  időpillanatbeli sorhossz.  $V'$  legyen az  $i$ -ik vakáció periódus vége,  $i = 1, \dots, \zeta$  esetén:

$$V' = \sum_{j=1}^i V_j \text{ és } I = V'_\zeta = \sum_{i=1}^{\zeta} V_i$$

$T_Z$  lesz a foglalt  $B$  állapot kezdete. Valamint legyen  $Z = X(T_Z)$  a foglalt állapot kezdeténél a sorhossz.  $T_i$  lesz az az időpillanat mikor a sorhossz lecsökken  $i$ -re.

A  $T_i$  időket jelöljük  $Z$ -vel és legyenek  $C$  részperiódusai.

$X(t)$  értékét az alábbi grafikonon tudjuk ábrázolni:



Miután vázoltuk a rendszerünket, számítsuk ki  $\zeta$  és  $Z$  eloszlását.

#### 4.1 A rendszerben lévő vakációk száma, a tétlen periódus és a foglalt periódus

Hogy meghatározzuk  $\zeta$ -t avagy a rendszerben lévő vakációk számát a tétlen állapotban, először figyeljük meg, hogy a  $\zeta \geq i$  esemény megegyezik azzal, hogy nincs beérkező igény

$V'_{i-1} = \sum_{k=1}^{i-1} V_k$  alatt. Nevezzük el  $A_k$ -nak, hogy nincs érkező igény  $V_k$  alatt.  $A_k^c$ -nak meg az

erre komplementes eseményt. Legyen  $L_K(s) = E(\exp(-sV_K))$  a  $V_K$  Laplace-Stieltjes transzformáltja. Ekkor  $i > 1$ -re  $\lambda$  paraméterrel:

$$P(\zeta=1) = P(A_1^c) = E[1 - \exp(-\lambda V_1)] = 1 - L_1(\lambda)$$

$$P(\zeta=i) = P(A_1^c) \prod_{k=1}^{i-1} P(A_k) = \prod_{k=1}^{i-1} L_k(\lambda)$$

$$P(\zeta \geq 1) = \prod_{k=1}^{i-1} P(A_k) = \prod_{k=1}^{i-1} L_k(\lambda)$$

A  $\prod_{k=a}^b L_k(\lambda)$  értéke 1, ha  $b < a$ . Ezek alapján:

$$E[\zeta] = \sum_{i=1}^{\infty} iP(\zeta=i) = \sum_{i=1}^{\infty} P(\zeta \geq i) = \sum_{i=1}^{\infty} \prod_{k=1}^{i-1} L_k(\lambda)$$

A tétlen periódus várható értéke:

$$E[I] = \sum_{i=1}^{\infty} E[V_i] \prod_{k=a}^b L_k(\lambda)$$

A foglalt periódus a  $Z$  részperiódusokból áll. Azaz leírhatjuk ebben a formában:  $B = \sum_{i=1}^Z B_i$

$B_1$  egy sima M/G/1 rendszer vakációk nélkül. A  $B_i$ -k eloszlása azonos. Azért:

$$E[B] = E[E[B \vee Z]] = E[ZE[B_1]] = E[Z]E[B_1]$$

Tudjuk, hogy  $\rho := \lambda E[\sigma]$  és  $E[B_1] = \frac{E[\sigma]}{1-\rho}$ , így  $E[B] = E[B_1] = \frac{\rho}{1-\rho}(E[I] + T_w)$

## 5.0 Hosszú farkú osztott processzoros (PS) rendszerek

### 5.1 M/G/1/PS rendszer

A kiszolgálás szabályára és struktúrájára vonatkozóan további mennyiségeket kell meghatározni. Az egyik a *befogadóképesség*, ami nem más, mint a várakozó sor maximális hossza. Ezt rendszerint  $K$ -val jelöljük, és értékét gyakran végtelennek tekintjük. Egy további jellemző a rendelkezésre álló *kiszolgáló állomások (csatornák) száma*. A *kiszolgálási elv* adja meg az igények kiszolgálásának sorrendjét. A leggyakrabban használt kiszolgálási elvek : FIFO (First In - First Out) - érkezési sorrendben; LIFO (Last In - First Out) - fordított sorrendben; SIRO (Service In Random Order) - véletlen sorrendben történő kiszolgálások. Ha a beérkező igényeket bizonyos csoportokba tartozás szerint meg lehet különböztetni, és a csoportok között *prioritást* lehet megállapítani, akkor ezen a prioritáson alapul a kiszolgálás sorrendje. Ez az egyik legalkalmasabb ütemezési elv, mivel így az igények közötti fontossági sorrendet felállítva történik a kiszolgálás. A prioritásos sorbanállási elvnek két fő típusa van: *abszolút* és *relatív*. Az előbbi azt jelenti, hogy ha egy igény kiszolgálása folyamatban van, és érkezik egy magasabb prioritású igény, akkor az éppen kiszolgálás alatt álló folyamat kiszolgálása megszakad, és újra beáll a várakozási sorba, míg a magasabb prioritású kiszolgálása elkezdődik. Relatív prioritásnál a kiszolgálás nem szakad meg, hanem annak végeztével a kiszolgáló a legmagasabb prioritással rendelkező igénnyel folytatja munkáját. A számítógéprendszerek egyik leggyakoribb kiszolgálási elve a TS (Time Sharing) vagy más terminológiával RR (Round Robin) - *időosztásos diszciplína* -, ami a következőt jelenti. A kiszolgálásra várakozó jobok mindegyikét a CPU bizonyos  $q$ , ú.n. kvantum ideig szolgálja ki. Ha ennyi idő alatt a job kiszolgálása teljesen befejeződik, akkor távozik a CPU-tól, ha nem, akkor ismét beáll a várakozók sorába. Ez az elv azt eredményezi, hogy a rövid CPU időt igénylő jobok nem tartózkodnak sokáig a CPU-nál. Mivel ezt az elvet elég nehéz matematikailag modellezni, vagyis megfelelő sztochasztikus folyamatot konstruálni hozzá, ezért szívesebben foglalkozunk ezen elv határesetével, amikor is a  $q$  nagyon kicsi. Ekkor kapjuk az ú.n. PS (Processor Sharing) -*osztott processzoros*- kiszolgálási diszciplínát. Ekkor, ha  $\Delta t$  idő

alatt  $t$  job tartózkodik a CPU-nál, akkor mindegyik kiszolgálása folyik, de nem  $1$ , hanem  $1/k$  intenzitással. Vagyis  $\Delta t$  idő alatt mindegyik job igényelt kiszolgálási idejéből  $\Delta t/k$  egység telt el.

Az adatforgalom nagy változékonyságát mind az idő tartományában (a forgalom összefüggőségi kapcsolatai nem csengenek le exponenciális gyorsasággal, mint a beszédforgalom esetében hanem annál sokkal lassabban, hosszú idejű összefüggések is jelen vannak), mind a méret tartományában (a forgalmi egységek méretének eloszlása nem exponenciális, mint beszédforgalomnál, hanem a hosszúfarkú eloszlások a tipikusak, pl. a web által letöltött objektumok mérete). Ezen tényezők hatására új modellek és technikákat fejlesztettek ki. A hosszabb időskálájú összefüggőségi struktúrát statisztikusan leírhatjuk a *hosszú idejű összefüggőséggel* (long-range distributions, LRG) Az extrém méretbeli változékonyságot, pedig a végtelen szórású *hosszú farkú eloszlások* segítségével. (heavy-tailed distributions)

## **5.2 Hosszú farkú osztott processzoros (PS) rendszerek türelmetlenséggel és áteresztés irányítás Packet Network hálózatokban**

Ebben a fejezetben az M/G/1 osztott processzoros sort, hosszú farkú szolgáltatásokkal és türelmetlen igényekkel tanulmányozzuk. Feltételezzük, hogy a türelmetlenség függ a szolgáltatás igényének értékétől. A csökkentett szolgáltatás arány (reduced service rate – RSR) közelítés becslést ad az igény rendszerben való tartózkodási idejére, mikor a sor kapacitása véges vagy végtelen. Így lehetőségünk van megbecsülni az igény visszaérkezési valószínűségét hosszú idejű szolgáltatás alatt. Ezeket az eredményeket arra használjuk, hogy megvizsgáljuk az áteresztés irányítás hatását egy packet hálózaton. Az áteresztés ellenőrzés az egyszerre történő kapcsolódások számát figyeli. Hatalmas előnyt jelent a rendszer hatékonyságára nézve az áteresztés irányítás végrehajtása. Globálisan növeli azon igények hányadosát amelyek elvégezték a szolgáltatásukat.(azaz nem lesznek türelmetlenek). Ennek segítségével meg tudjuk adni a rendszer kapacitásának kritériumait.

Az utóbbi időben a szolgáltatás minősége (quality of service QoS) problémák kevésbé

kritikusakká váltak a packet hálózatok gerincére nézve, az átméretezésnek köszönhetően. De a QoS még mindig a legkomolyabb probléma a hálózat hozzáférés megtervezéséhez. A packet áramlást három paraméterrel azonosítjuk. A  $(\sigma, \rho, \pi)$  hármassal, ami a hálózat kalkulus alapjaként is ismerünk. ( $\rho$ , és  $\pi$  határozzák meg a maximum és átlag értékeket, míg  $\sigma$  a bucket méret) Ezeket a paramétereket használjuk az áteresztés irányítást implementáló algoritmusnál is. Ezen paraméterek meghatározása a felhasználók számára egy nehéz feladat.

### 5.2.1 Az áteresztés irányítás alapértékei

A PS technika alkalmazásánál egy fájl átküldésének ideje érzékeny az összeköttetés terheltségére. Átlagosan az átviteli idő  $x$  méret mellett  $x/c(1 - \rho)$ , ahol  $\rho$  az átlagos terheltség és  $c$  az összeköttetés kapacitása. Mikor  $\rho$  1 közelebb, akkor ez az időmennyiség nagyon nagy szám, ami sokszor a felhasználókat a szolgáltatás megszakításához vezeti.

Az áteresztés irányítás előnye, hogy növeli a sikeres átküldések számát valamint lehetővé teszi, hogy bizonyos áramlásokat visszautasítson bármilyen processzor folyamat előtt, ezzel csökkenti a felhasználók türelmetlenségét.

A felhasználók  $\lambda$  paraméterrel érkeznek és hosszú-farkú kiszolgálási időre van szükségük. Jelöljük ezt az értéket  $B$ -vel. Valamint jelöljük a türelmetlenség idejének értékét  $C$ -vel. Az igényeket PS szisztéma szerint szolgáljuk ki és egy igény elhagyja a rendszert ha kiszolgálták vagy eléri a  $C$  értéket. Azon felhasználók akik nagyobb filét küldenek, hajlandóak több időt várni és nagyobb a türelmetlenségi értékük, mint azon felhasználóknak akik kisebb filét küldenek.  $C$ , a  $B$  kiszolgálási időre vett függvénye egy felhasználóra nézve egy nem csökkenő konkáv függvény, ha tekintjük, hogy a felhasználók egyre türelmetlenebbek ahogy a fájlátvitel ideje nő. Legyen  $C = \theta B$  és  $\theta > 1$ . A sor várakozási kapacitása legyen  $N$ . Mikor az aktív folyamatok száma eléri az  $N$ -t, akkor a folyamatokat visszautasítja a rendszer. Ennek a valószínűsége legyen  $P_N$ . Valamint a türelmetlenség miatt távozó felhasználók aránya legyen  $K_N$ . Az összveszteség valószínűség így  $\pi_N = K_N + P_N$ . Így  $1 - \pi_N$  a kiszolgált felhasználók arányszáma. Mikor a puffer kapacitása végtelen, akkor a visszautasítás valószínűség értéke közelítőleg

egyenlő a nullával. ( $P_{\infty} = 0$ )

Az áteresztés irányítás azt mondják akkor működik effektíven, ha  $N$  választásánál  $K_N \ll P_N$ .

Azaz a türelmetlenség kisebb értékű, mint a visszautasítások. Így kevesebb felhasználó tartózkodhat a rendszerben, de ha már egyszer bekerültek, akkor türelmetlenség miatt nem kell távozniuk.

### 5.2.2 Az RSR megközelítés

Feltételezzük, hogy az azonosított felhasználók folyamatosan a rendszerben vannak, az  $(L(u); u \geq 0)$  függvény jelenti a rendszerben lévő felhasználók számát az  $u$  időpontban. Így a  $x$  ideig fogadott szolgáltatások száma egy azonosított ügyfél részéről:

$$R(x) = \int_0^x \frac{1}{1+L(u)} du$$

Legyen  $B$  néhány  $L(u)$ -tól független random változó és  $V = \inf\{y \geq 0; R(y) \geq B\}$ , akkor  $V$  az az időmennyiség ami megadja egy  $0$  időpontban beérkező ügyfélre vonatkozó a szolgáltatás időhosszát. Ha  $V$  nem türelmetlen, akkor  $V$  a sorban való tartózkodás idejének értéke.

Vegyük a következő állításokat:

$$R(x)/x \rightarrow \gamma = 1 - (1 - b_N)\rho,$$

Ahol  $b_N$  a visszautasítások valószínűsége a PS sorban egy folyamatosan tartózkodó ügyfél esetén és ha a sor kapacitását  $N$ -nek adjuk meg.

Az RSR reláció helytálló, ha:

- $B$  pontos indexe  $\nu > 1$ .
- $R(x)/x \rightarrow \gamma$  és  $x \rightarrow \infty$  ha  $0 < \gamma < 1$
- Létezik egy pozitív, véged konstans amire:  $R(x) \leq x/K = o(P(B > x))$

Ezek akkor igazak ha a puffer méret véges. (érdemes  $K$ -t választani pufferméretnek)

### 5.2.3 Szimulációs eredmények

Az alábbi diagramoknál megtekinthetünk a felhasználók eloszlását egy M/G/1 PS sorban, ahol a türelmetlenségi együttható  $\theta = 10$ , különböző  $\rho$  terheltségi értékek mellett és a kiszolgálási idők exponenciális vagy Pareto eloszlásúak és  $\beta = 1$ ,  $\alpha = 1.25$ .

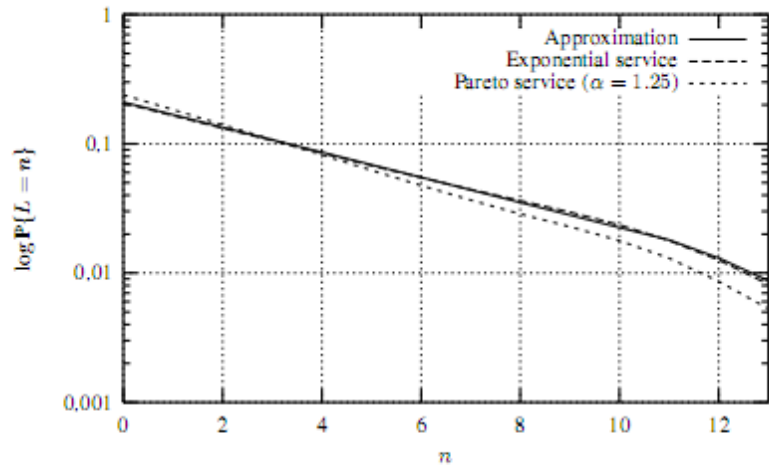


Fig. 1. Approximation vs. simulation results for exponential and Pareto ( $\nu = 1.25$ ) service times for  $\rho = 80\%$ ,  $\theta = 10$ ,

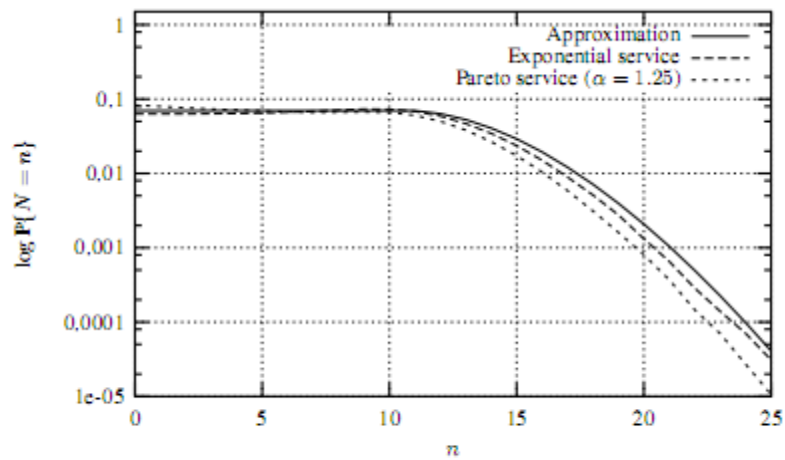


Fig. 2. Approximation vs. simulation results for exponential and Pareto ( $\nu = 1.25$ ) service times for  $\rho = 100\%$ ,  $\theta = 10$ ,

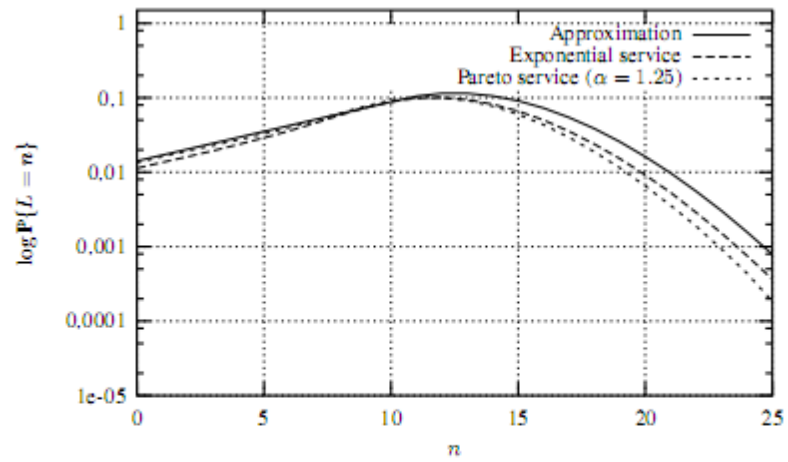


Fig. 3. Approximation vs. simulation results for exponential and Pareto ( $\nu = 1.25$ ) service times for  $\rho = 120\%$ ,  $\theta = 10$ .

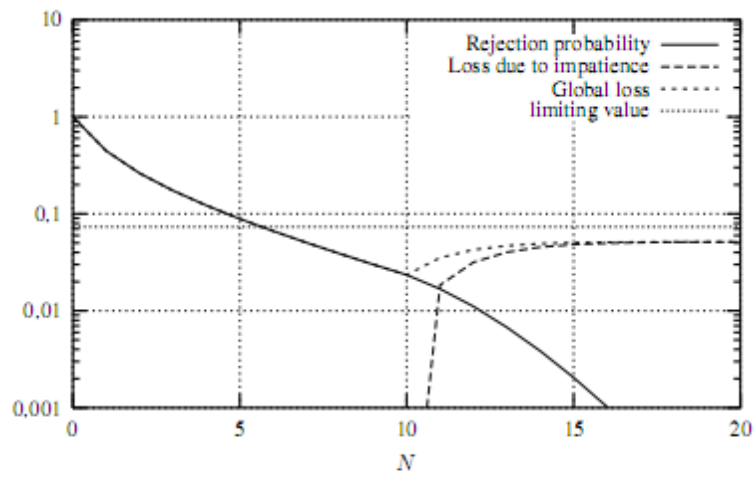


Fig. 4. Loss due impatience and to rejection for  $\rho = 80\%$  and  $\theta = 10$ .

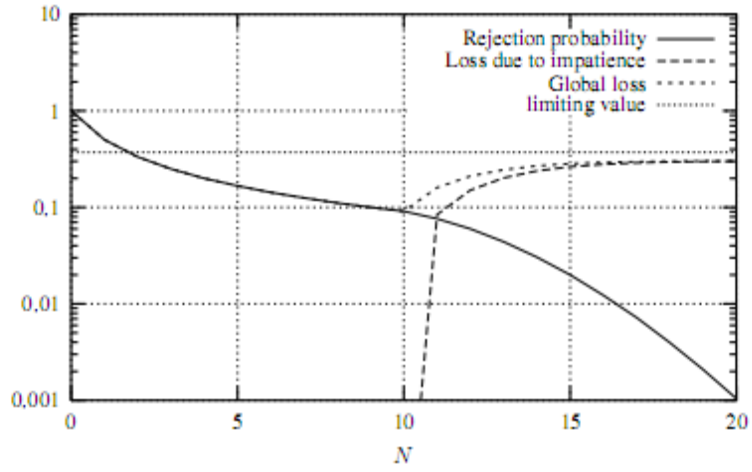


Fig. 5. Loss due impatience and to rejection for  $\rho = 100\%$  and  $\theta = 10$ .

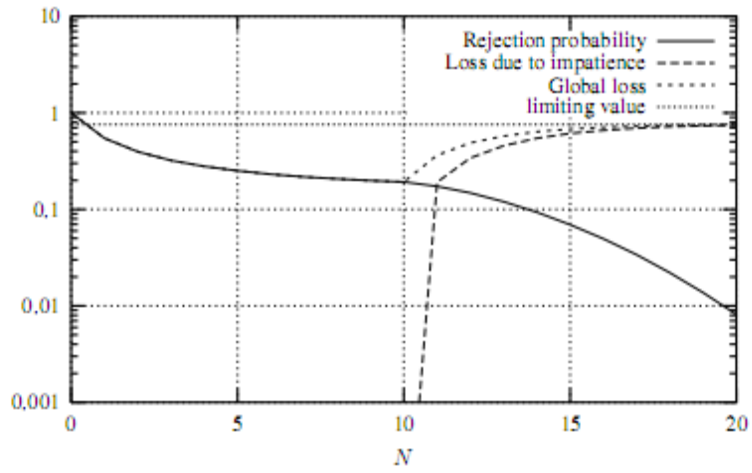


Fig. 6. Loss due impatience and to rejection for  $\rho = 120\%$  and  $\theta = 10$ .

Az alábbi

táblázatban nyomon követhetjük a Pareto és exponenciális kiszolgálási idők mellett a  $K_N$  eloszlását ha  $N$  végtelen.

	$\rho = 80\%$		$\rho = 100\%$		$\rho = 120\%$	
	$\theta = 3$	$\theta = 10$	$\theta = 3$	$\theta = 10$	$\theta = 3$	$\theta = 10$
Pareto ( $\nu = 1.25$ )	0.3642	7.1014e-2	0.5574	.37630	0.7036	0.7158
Exponential	0.3856	7.7067e-2	0.5681	0.3718	0.7182	0.7279
Approximation	0.3016	7.3926e-2	0.641227	0.3720	0.9164	0.8470
$P_{[s]}$	0.1734	2.3493e-2	0.2500	0.0910	0.3219	0.1925

#### 5.2.4 M/G/s PS sor esetén

A PS sor több kiszolgáló esetén a következőképpen néz ki. A felhasználók száma a rendszerben  $\mathbf{u}$  időpillanatban  $\mathbf{Q}(\mathbf{u})$ . Ha  $\mathbf{Q}(\mathbf{u}) \geq s$ , akkor minden felhasználó  $s/\mathbf{Q}(\mathbf{u})$  ráta szerint lesz kiszolgálva. Egyébként a felhasználók 1-es kiszolgálási értéket kapnak. Ezen modell célja, hogy a maximum fájlátvitel arányai lényegesen kisebbek, mint a teljes rendszer kapacitása.

## 6.0 Konklúzió

A dolgozat célja az M/G/1 sorbanállási rendszer ismertetése, összefoglalása valamint példákön keresztüli bemutatása volt. A dolgozatban lévő számítások hálózatok megtervezéséhez adhat nagy segítséget, ha szeretnénk optimális kihasználtságú hálózatokat tervezni.

A mellékelt programban lévő számolások hasznosak lehetnek ha konkrét értékeket szeretnék a tervezésnél kipróbálni, segítségével könnyebben kialakítható a megfelelő sorbanállási modell.

Az utolsó két fejezetben konkrét példákön keresztül láthatja az olvasó az M/G/1 rendszer használatának előnyeit. Ez a két példa csak kettő a sok-sok lehetőség közül, amiket be lehetne mutatni ebben a témában. Bemutatásuk célja az volt, hogy lássuk gyakorlatban is a sorbanállás elmélettel való vizsgálatok és számítások fontosságát.

A dolgozat célja – az M/G/1 rendszer ismertetése és összefoglalása – sikeres volt, valamint a két példa is érdekes és hasznos információkkal látta el az olvasót. Így összességében úgy érzem, sikeres lett a célkitűzés megvalósítása.

## **7.0 Köszönetnyilvánítás**

Ezúton szeretném megköszönni dr. Sztrik János tanárúrnak, a témavezetőmnek, hogy segített a dolgozat elkészítésében, és a segítőkész hozzáállását az egész diplomamunkával kapcsolatban. Valamint a családomnak a támogatást, és a megértő hozzáállást az egyetemen töltött éveim alatt.

## 9.0 Irodalomjegyzék

[1] dr. Sztrik János:

Kulcs a sorbanállási elmélethez és alkalmazásaihoz

Debrecen, 2000-2001

[2] dr. Sztrik János:

Bevezetés a sorbanállási elméletbe és alkalmazásaiba

Debrecen, 2004.

[3] J. Virtamo:

Queueing Course, from Finland. Complete lecture notes, by J. Virtamo.

2005.

[4] dr. Molnár Sándor:

Forgalomelmélet

Budapest, 2005

[5] Jacqueline Boyer, Fabrice Guillemin, Philippe Robert, Bert Zwart:

Heavy tailed M/G/1-PS queues with impatience and admission control in packet networks

IEEE INFOCOM, 2003

[6] Sara Alouf, Eitan Altman and Amar P. Azad:

Analysis of an M/G/1 Queue with Repeated Inhomogeneous Vacations Application to IEEE

802.16e Power Saving Mechanism

INRIA Sophia Antipolis, B.P. 93, 06902, Sophia Antipolis Cedex, France

## 10.0 Függelék:

Little formula:

Az  $N = \lambda W$  összefüggést Little-formulának nevezzük, ahol  $N$  a rendszerben tartózkodó igények átlagos száma,  $\lambda$  az igények átlagos érkezési intenzitása és  $W$  az átlagos várakozási idő a rendszerben. A Little formula érvényes minden stacionárius rendszerre ahol a rendszerben igények nem születnek és nem vesznek el.

PASTA összefüggés:

Poisson (érkezési folyamatok esetén exponenciálisak az érkezések közötti idők eloszlása), állandósult állapotban egy tetszőleges pillanatban az aktív kapcsolatok számának eloszlása egyenlő és az érkezés pillanatában lévő kapcsolatok számának eloszlásával. Ennek az összefüggésnek PASTA (Poisson arrivals see time averages) a neve mert a valószínűség egyenlő az aktív kapcsolatok átlagos időhányadával ha hosszú időn át vizsgáljuk a rendszert.

Markov tulajdonság:

Exponenciális érkezések közötti idők eloszlása esetén egy tetszőleges időpillanat után a hátralévő idő eloszlása esetén egy tetszőleges időpillanat után a hátralévő idő eloszlása is exponenciális ugyanazzal a paraméterrel, az a modell, melynél az érkezések közötti idő és a kiszolgálási idő is exponenciális Markovi modell. Ettől eltérő esetekben a modellt nem Markovi modellnek nevezzük.