

# A heterogén forgalom hatása a Proxy Cache szerverek hatékonyságára

## The impact of heterogeneous traffic on the performance of Proxy Cache servers

\*Bérczes Tamás, Sztrik János

Informatikai Kar, Debreceni Egyetem, Magyarország  
{berczes.tamas, sztrik.janos}@inf.unideb.hu

### Összefoglaló

Az Internet használhatóságának egyik elengedhetetlen feltétele, hogy a különböző igények, lekérdezések válaszára a forgalom bővülésétől függetlenül viszonylag kicsi maradjon. Jelen dolgozat keretében a már korábban használt Proxy Cache szerverek hatékonyság vizsgálatára felállított nyitott Jackson sorbanállási modellt módosítjuk, hogy vizsgálni tudjuk a heterogén forgalom hatását a válaszidőkre. A kliens által igényelt fájlokat méretük alapján két csoportba sorolva, megvizsgáljuk, hogyan hat a válaszidőre a Proxy Cache szerver jelenléte, különböző rendszerparamétereket használva.

**Kulcsszavak:** Sorbanállási rendszerek, Proxy Cache Szerver

### Abstract

The focus of the paper is to examine the performance behavior of Proxy Cache servers when we use heterogeneous traffic. We describe the multi-class queuing network model of the Proxy Cache server, where we separate the requests in two classes by virtue of their size. If the size of the requested document is greater than average we put it into class *a*. In the opposite case, when the size of the requested file is small we put it into class *b*. We have calculated the overall response time with and without a Proxy Cache server. We analyzed how various factors affect the performance of a Proxy Cache server when we use heterogeneous traffic. In general when the arrival rate of requests increases, then the response time increases as well regardless of the existence of a Proxy Cache server. Increasing the percentage of the class *a* the response time increases too. When we use a higher percentage of the class *a* and we use a high arrival rate, then the response time gap is more significant between the cases with and without a Proxy Cache server. Using a low percentage of class *a* files, a low arrival rate and low cache hit rate probability we get higher response time in presence of a Proxy Cache server.

Several numerical examples illustrate the effect of arrival, external arrival rate and the file size of class *a* and class *b* on the mean response times.

**Keywords:** Queueing Network, Proxy Cache Server

## 1. Bevezetés

Napjainkban az egyik leginkább közkedvelt információszerezési lehetőség az internet használata. Az internet gyors és egyszerű lehetőséget biztosít több ezer Webszerver adatainak a megismerésére, letöltésére. Az internet használata az elmúlt években rohamosan növekedett. A felhasználók száma a 2001-es 474 millióról 2002-re 590 millióra növekedett. 2006-ra az internetet használók száma elérte a 948 milliót. Figyelembe véve, hogy 1996-ban mindösszesen 40 millióan használták az internetet, a növekedés üteme igen jelentős. A felhasználók számának

\*A publikáció elkészítését a TÁMOP 4.2.1./B-09/1/KONV-2010-0007 számú projekt támogatta. A projekt az Új Magyarország Fejlesztési Terven keresztül az Európai Unió támogatásával, az Európai Regionális Fejlesztési Alap és az Európai Szociális Alap társfinanszírozásával valósult meg.

növekedésével párhuzamosan növekedett az internet forgalma is. Ennek hatására egyre nagyobb igény mutatkozik a színvonalas és gyors internet elérésre és kiszolgálásra. Az információ-keresés és letöltés közben a válasz a távoli Web szervertől a kliens gépéig gyakran igen sok időt vesz igénybe. Egy tartalom többszöri és egyidejű elérése miatt a válaszidők növekednek, ezért indokolt a tartalmak felhasználók környékén való tárolása, amely által a késleltetés csökkenthető. Ennek egyik megoldási lehetősége a böngésző szoftverbe való implementálás ([1]). Ebben az esetben a tárolt adatokhoz azonban csak egy személy férhet hozzá. Egy másik lehetőség Proxy Cache szerver használata.

A felhasználó szemszögéből nézve lényegtelen, hogy az általa keresett fájl fizikailag hol található: egy Proxy Cache szerveren valahol a munkahelyének belső hálózatán vagy a világ túlsó felén egy távoli Web szerveren. A keresett dokumentum érkezik a távoli Web szervertől vagy a Proxy Cache szervertől. Kliens oldalról nézve a Proxy Cache szerver funkciója ugyanaz mint egy Web szervernek, valamint a Web szerver felől nézve a Proxy Cache szerver ugyanúgy viselkedik, mint egy kliens.

Feltételezhető, hogy egy Proxy Cache szerver beüzemelése egy cég belső hálózata és az internet közé, kisebb sávszélesség igényt valamint kisebb válaszidőket eredményezhet [5]. Így a vállalatok több felhasználót kapcsolhatnak ugyanakkora sávszélességre, mivel a Proxy Cache szerver redundánsan tárolja az adatokat, több felhasználó számára.

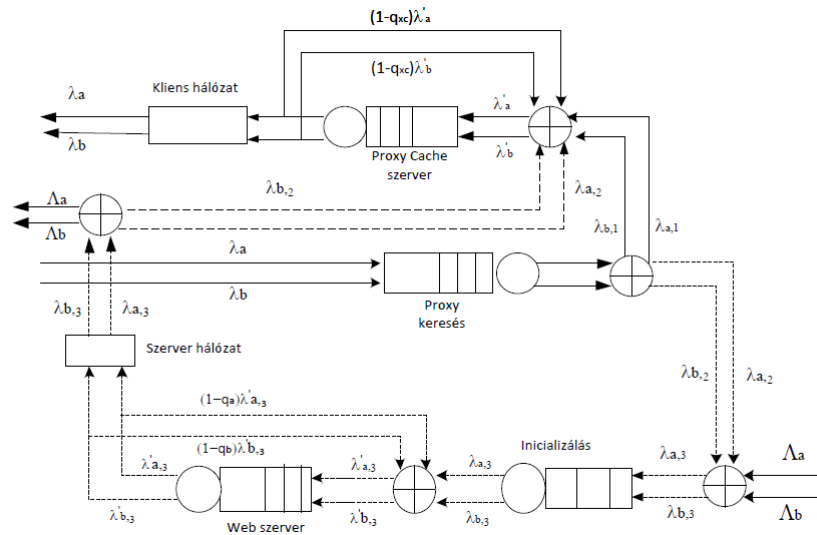
Jelen cikkben megvizsgáltam, milyen hatással van a heterogén forgalom a Proxy Cache szerver hatékonyságára. Ebben az esetben a keresett fájlokat a méretük alapján két osztályba soroljuk. Ebben az esetben a válaszidők kiszámításához külön kell vizsgálni a két csoportba tartozó kérések válaszidejeit, majd ezek segítségével kaphatjuk meg egy tetszőleges igény átlagos válaszidejét.

## 2. Proxy Cache szerver heterogén modellje

Proxy Cache szervert használva, ha egy fájlt le akarunk tölteni egy távoli Web szerverről, először meg kell vizsgálni, hogy a keresett fájl megtalálható-e a Proxy Cache szerveren. Ennek a valószínűségét  $p$ -vel jelöljük. Amennyiben a keresett dokumentum megtalálható a Proxy Cache szerveren, egy másolat a fájlról azonnal továbbítódik a felhasználónak. Amennyiben a dokumentum nem található meg a Proxy Cache szerveren, az igény továbbítódik a távoli Web szerverhez. A dokumentum a Web szerverről először a Proxy Cache szerverre érkezik vissza, ahonnan egy másolat a fájlról azonnal a felhasználóhoz kerül. Az eredeti példány tárolódik a Proxy Cache szerveren, így a későbbiekben elérhető lesz a felhasználók számára. A Proxy Cache szerver hatékonysága a következő tényezőktől függ:

- találati arány (a kért dokumentum milyen valószínűséggel található meg a Proxy Cache szerveren)
- Proxy Cache szerver karakterisztikája - sebessége
- kliens oldali sávszélesség
- szerver oldali sávszélesség
- a belső és külső igények érkezési intenzitása
- Web szerver karakterisztikája - sebessége

A kliensek által keresett fájlokat két osztályba soroljuk a méretük alapján. Amennyiben a fájl mérete az átlagosnál nagyobb az  $a$  osztályba soroljuk, míg ellenkező esetben amikor a fájl mérete kicsi "normál" fájlról beszélünk és a  $b$  osztályba soroljuk. Mindkét osztályba tartozó igény esetén először megvizsgáljuk, hogy a fájl megtalálható-e a Proxy Cache szerveren vagy sem. Ezt a találati valószínűséget  $p_a$  illetve  $p_b$  -vel jelöljük az  $a$  illetve a  $b$  osztályhoz tartozó fájlok esetén. Amennyiben a keresett fájl megtalálható a Proxy Cache szerveren, akkor mindkét osztály esetén a fájl egy másolata azonnal továbbítódik a klienshez. Ellenkező esetben, amikor is a fájl nem található meg a Proxy Cache szerveren az igény továbbítódik a távoli Web szerverhez függetlenül az osztályától. Miután az igényelt fájl visszaérkezik a Proxy Cache szerverhez egy másolat továbbítódik a klienshez. Az 1 ábra mutatja egy belső fájl lehetséges útját az igény indulásától egészen a fájl klienshez való megérkezéséig. Az ábrán az  $a$  illetve  $b$  index jelöli, hogy a keresett fájl az  $a$  vagy a  $b$  osztályhoz tartozik. Az ábrán és a cikkben használt jelölések megtalálhatóak a 4 táblázatban.



1. ábra. Proxy Cache szerver heterogén forgalmi modellje

Feltételezzük, hogy a belső  $a$  osztályhoz tartozó igények a Proxy Cache szerverhez  $\lambda_a$ , míg a  $b$  osztályhoz tartozó igények  $\lambda_b$  paraméterű Poisson-folyamat szerint érkeznek, valamint a Web szerverhez kívülről érkező igények  $\Lambda_a$  illetve  $\Lambda_b$  paraméterű Poisson-folyamat alapján érkeznek az  $a$  illetve  $b$  osztályhoz tartozó igények esetén.

Az egyenes vonal ( $\lambda_{a,1}$  ill.  $\lambda_{b,1}$ ) reprezentálja azt az esetet amikor a keresett fájl megtalálható a Proxy Cache szerveren. Szaggatott vonallal rajzolva jelöltük ( $\lambda_{a,2}$  ill.  $\lambda_{b,2}$ ) azon igények útját, melyek nem találhatóak meg a Proxy Cache szerveren, így ezek az igények továbbítódnak a távoli Web szerverhez. A  $\lambda_{a,1}$  és  $\lambda_{a,2}$  valamint  $\lambda_{b,1}$  és  $\lambda_{b,2}$  intenzitások értékei:

$$\lambda_{a,1} = p_a * \lambda_a \text{ és } \lambda_{b,1} = p_b * \lambda_b, \tag{1}$$

$$\lambda_{a,2} = (1 - p_a) * \lambda_a \text{ és } \lambda_{b,2} = (1 - p_b) * \lambda_b. \tag{2}$$

A Web szerverhez érkező igények teljes intenzitása a Web szerver felé továbbított belső igények illetve a külső igények intenzitásának az összege, azaz

$$\lambda_{a,3} = \Lambda_a + \lambda_{a,2} \text{ és } \lambda_{b,3} = \Lambda_b + \lambda_{b,2}. \tag{3}$$

A Web szerverhez érkező igényeknek át kell esniük egy egyszeri inicializálási folyamaton, melyet a 1 ábrán az "Inicializálás" csomópont szemléltet. Az egyszeri inicializáláshoz szükséges idő mindkét osztályhoz tartozó fájlok esetén (lásd a [2]. cikkben):

$$\frac{1}{I_s - (\lambda_{a,3} + \lambda_{b,3})}. \tag{4}$$

A Web szerver valamint a Proxy Cache szerver karakterisztikáját meghatározó paraméterek  $B_s, Y_s, R_s$  valamint  $B_{xc}, Y_{xc}, R_{xc}$  rendre a szerver kimenő puffere, a statikus szerver idő valamint a dinamikus szerver arány (lásd a [2]. cikkben), alapján a Web szerver és a Proxy Cache szerver kiszolgálási intenzitása

$$\mu_{Web} = \frac{1}{Y_s + \frac{B_s}{R_s}}, \tag{5}$$

$$\mu_{PCS} = \frac{1}{Y_{xc} + \frac{B_{xc}}{R_{xc}}}. \tag{6}$$

Ha a keresett fájl nagyobb mint a Web szerver kimenő puffere, akkor egy visszacsatolási ciklus kezdődik, mely addig tart míg a teljes fájl kiszolgálása be nem fejeződik. Legyen  $q_a$  illetve  $q_b$  annak a valószínűsége, hogy keresett  $a$  illetve  $b$  osztályhoz tartozó fájl egyből sikerül továbbítani, ahol

$$q_a = \min \left( 1, \frac{B_s}{F_a} \right) \quad (7)$$

illetve

$$q_b = \min \left( 1, \frac{B_s}{F_b} \right). \quad (8)$$

Teljesen hasonlóan modellezhető a Proxy Cache szerver is, ahol a távozó folyamat visszacsatolásának a valószínűsége  $1 - q_{a,xc}$  illetve  $1 - q_{b,xc}$ , ahol

$$q_{a,xc} = \min \left( 1, \frac{B_{xc}}{F_a} \right) \quad (9)$$

az  $a$  osztályhoz tartozó fájl esetén, illetve

$$q_{b,xc} = \min \left( 1, \frac{B_{xc}}{F_b} \right) \quad (10)$$

a  $b$  osztályhoz tartozó fájl esetén.

A belső  $a$  osztályhoz tartozó igények válaszidejét  $T_a^{xc}$ -vel valamint a belső  $b$  osztályhoz tartozó igények válaszidejét  $T_b^{xc}$ -vel jelöljük, melyeket a következő képletek határoznak meg:

$$\begin{aligned} T_a^{xc} = & \frac{1}{\frac{1}{I_{xc}} - (\lambda_a + \lambda_b)} \\ & + p_a * \left\{ \frac{\frac{1}{q_{a,xc}} * (Y_{xc} + \frac{B_{xc}}{R_{xc}})}{1 - \sum_{j=a}^b \frac{\lambda_j}{q_j} (Y_{xc} + \frac{B_{xc}}{R_{xc}})} + \frac{F_a}{N_c} \right\} \\ & + (1 - p_a) * \left\{ \frac{1}{\frac{1}{I_s} - (\lambda_{a,3} + \lambda_{b,3})} + \frac{\frac{1}{q_a} * (Y_s + \frac{B_s}{R_s})}{1 - \sum_{j=a}^b \frac{\lambda_{j,3}}{q_j} (Y_s + \frac{B_s}{R_s})} + \frac{F_a}{N_s} \right. \\ & \left. + \frac{\frac{1}{q_{a,xc}} * (Y_{xc} + \frac{B_{xc}}{R_{xc}})}{1 - \sum_{j=a}^b \frac{\lambda_j}{q_{j,xc}} (Y_{xc} + \frac{B_{xc}}{R_{xc}})} + \frac{F_a}{N_c} \right\}, \end{aligned} \quad (11)$$

és

$$\begin{aligned} T_b^{xc} = & \frac{1}{\frac{1}{I_{xc}} - (\lambda_a + \lambda_b)} \\ & + p_b * \left\{ \frac{\frac{1}{q_{b,xc}} * (Y_{xc} + \frac{B_{xc}}{R_{xc}})}{1 - \sum_{j=a}^b \frac{\lambda_b}{q_{b,xc}} (Y_{xc} + \frac{B_{xc}}{R_{xc}})} + \frac{F_b}{N_c} \right\} \\ & + (1 - p_b) * \left\{ \frac{1}{\frac{1}{I_s} - (\lambda_{a,3} + \lambda_{b,3})} + \frac{\frac{1}{q_b} * (Y_s + \frac{B_s}{R_s})}{1 - \sum_{j=a}^b \frac{\lambda_{j,3}}{q_j} (Y_s + \frac{B_s}{R_s})} + \frac{F_b}{N_s} \right. \\ & \left. + \frac{\frac{1}{q_{b,xc}} * (Y_{xc} + \frac{B_{xc}}{R_{xc}})}{1 - \sum_{j=a}^b \frac{\lambda_b}{q_{b,xc}} (Y_{xc} + \frac{B_{xc}}{R_{xc}})} + \frac{F_b}{N_c} \right\}. \end{aligned} \quad (12)$$

Így a teljes válaszidő:

$$T_{xc} = \frac{\lambda_a}{\lambda_a + \lambda_b} * T_a^{xc} + \frac{\lambda_b}{\lambda_a + \lambda_b} * T_b^{xc}. \quad (13)$$

A  $T_a^{xc}$  az  $a$  osztályhoz tartozó fájlok esetében egy belső igény átlagos válaszideje. Ennek a kiszámításához a hálózati modellünket három részhálózatra bontjuk. Ennek megfelelően a  $T_a^{xc}$  válaszidő három részből tevődik

össze. Az első rész annak az időtartama, míg eldől, hogy a keresett  $a$  osztályú fájl megtalálható-e a Proxy Cache szerveren vagy sem. (Lásd [2].)

A képlet második tagja annak a válaszideje, amikor a keresett fájl megtalálható a Proxy Cache szerveren, melynek sebessége, azaz kiszolgálási intenzitása a szerverkarakterisztika alapján:  $Y_{xc} + \frac{B_{xc}}{R_{xc}}$ . Ennek az esetnek a valószínűsége  $p_a$ . A második tag szintén két részből tevődik össze. Az első a Proxy Cache szervernél eltöltött idő, ahol a szerverhez érkező  $a$  osztályú fájlok érkezési intenzitása  $\lambda'_a = \frac{\lambda_a}{q_{a,xc}}$ . A képletrész második tagja, pedig a kliens hálózaton való áthaladási idő, mely a [2] alapján  $\frac{F_a}{N_c}$ . A képlet harmadik tagja azt az esetet írja le, amikor a fájl nem található meg a Proxy Cache szerveren, ezért az igény továbbítódik a távoli Web szerverhez. Ennek az esetnek a valószínűsége  $1 - p_a$ . A képlet ezen része további öt tagból áll. Az első az úgynevezett egyszerű inicializálási idő. A második tag az igény Web szervernél eltöltött ideje, ahol a Web szerverhez érkező igények intenzitása  $\lambda'_{a,3} = \frac{\lambda_{a,3}}{q_a}$ . A harmadik és az ötödik tag a fájlnak a szerver illetve kliens hálózaton való átjutáshoz szükséges "utazási" idő. A negyedik tag a Proxy Cache szerverhez visszaérkező igény kliens felé továbbításának az ideje.

A (12) képlet egy  $b$  osztályhoz tartozó belső igény válaszidejét jelöli. Amennyiben nem használunk Proxy Cache szert, akkor a keresett válaszidők a következőképpen alakulnak:

$$T_a = \frac{1}{\frac{1}{I_s} - ((\lambda_a + \Lambda_a) + (\lambda_b + \Lambda_b))} + \frac{\frac{1}{q_a} * (Y_s + \frac{B_s}{R_s})}{1 - \sum_{j=a}^b \frac{(\lambda_j + \Lambda_j)}{q_j} (Y_s + \frac{B_s}{R_s})} + \frac{F_a}{N_s} + \frac{F_a}{N_c}, \quad (14)$$

és

$$T_b = \frac{1}{\frac{1}{I_s} - ((\lambda_a + \Lambda_a) + (\lambda_b + \Lambda_b))} + \frac{\frac{1}{q_b} * (Y_s + \frac{B_s}{R_s})}{1 - \sum_{j=a}^b \frac{(\lambda_j + \Lambda_j)}{q_j} (Y_s + \frac{B_s}{R_s})} + \frac{F_b}{N_s} + \frac{F_b}{N_c}. \quad (15)$$

Így egy belső igény válaszideje Proxy Cache szerver nélkül:

$$T = \frac{\lambda_a}{\lambda_a + \lambda_b} * T_a + \frac{\lambda_b}{\lambda_a + \lambda_b} * T_b. \quad (16)$$

Megvizsgálva a (11)-(16) képleteket látható, hogy azokban az esetekben amikor valamelyik nevező nullához közelít a válaszidő a végtelenhez tart. Legyen  $\lambda_b/\lambda_a = \Lambda_b/\Lambda_a = m$  a  $b$  illetve  $a$  osztályhoz tartozó igények érkezési intenzitásának a hányadosa. Így a válaszidő a végtelenhez közelít, amennyiben a lenti egyenletek közül az egyik teljesül.

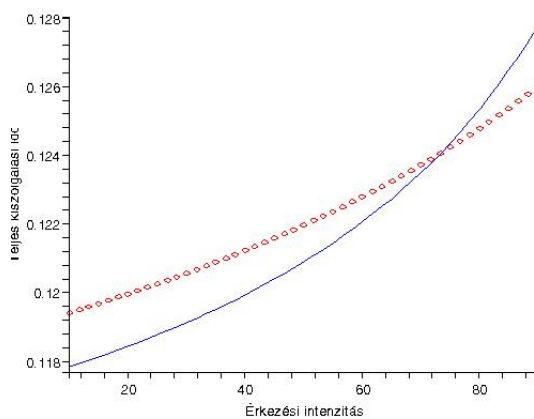
$$\begin{aligned} \lambda &= \frac{1}{I_{xc}}, \\ \lambda_a &= \frac{q_{a,xc}q_{b,xc}R_{xc}}{(q_{b,xc}+mq_{a,xc})(Y_{xc}R_{xc}+B_{xc})}, \\ \lambda_b &= \frac{mq_{a,xc}q_{b,xc}R_{xc}}{(q_{b,xc}+mq_{a,xc})(Y_{xc}R_{xc}+B_{xc})}, \\ \lambda_{a,3} + \lambda_{b,3} &= \frac{1}{I_s}, \\ \lambda_{a,3} &= \frac{q_a q_b R_s}{(q_b + m q_a)(Y_s R_s + B_s)}, \\ \lambda_{b,3} &= \frac{m q_a q_b R_s}{(q_b + m q_a)(Y_s R_s + B_s)}, \\ \lambda + \Lambda &= \frac{1}{I_s}, \\ \lambda_a + \Lambda_a &= \frac{q_a q_b R_s}{(q_b + m q_a)(Y_s R_s + B_s)}, \\ \lambda_b + \Lambda_b &= \frac{m q_a q_b R_s}{(q_b + m q_a)(Y_s R_s + B_s)} \end{aligned}$$

### 3. Numerikus eredmények

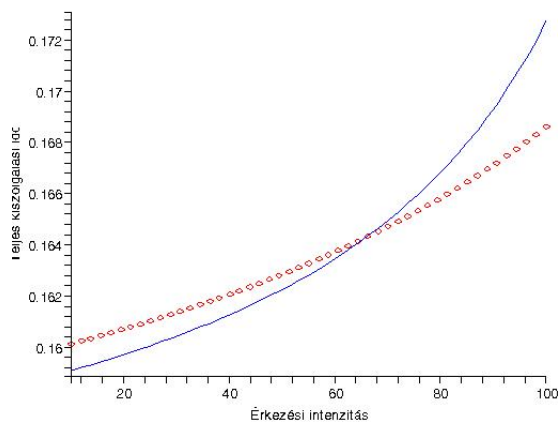
A következőekben vizsgált numerikus eredményekhez a használt szerver paraméterek a korábbi fejezetekben használtakkal megegyeznek. A számításokhoz a Web és Proxy Cache szerver paraméterek értékei [7] alapján:  $I_s = I_{xc} = 0.004$  másodperc,  $B_s = B_{xc} = 2000$  byte,  $Y_s = Y_{xc} = 0.000016$  másodperc,  $R_s = R_{xc} = 1.25$  Mbyte/másodperc,  $N_s = 1544$  Kbit/másodperc, valamint  $N_c = 128$  Kbit/másodperc. Az  $a$  illetve  $b$  osztályhoz tartozó fájlok méretét [7] alapján választottuk:  $F_a = 7000$  byte, valamint  $F_b = 1000$  byte. A fejezetben található grafikonokon pontozott vonallal ábrázoltuk a Proxy Cache szerveret tartalmazó esetet, valamint egyenes vonallal a Proxy Cache szerveret nem tartalmazó esetet.

#### 3.1. A belső igények érkezési intenzitásának hatása a válaszidőre

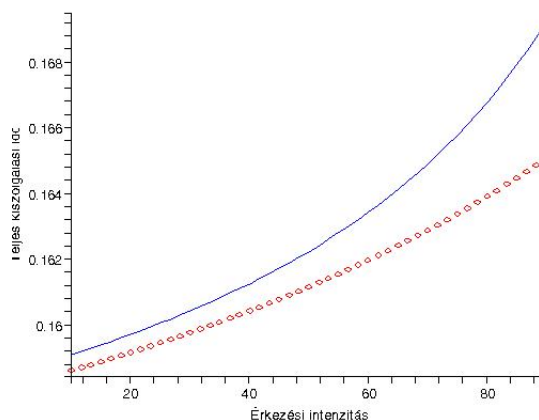
A 2 grafikonon a válaszidőt a belső igények érkezési intenzitásának függvényeként ábrázoltuk. Ebben az esetben az  $a$  osztályú fájlok aránya az összes igény között 10 %, a külső igények érkezési intenzitása 100 kérés/másodperc, a Proxy Cache szerveren a találati valószínűségek rendre  $p_a = p_b = 0.25$ . A két osztályhoz tartozó fájl méretek pedig  $F_a = 7000$  byte valamint  $F_b = 1000$  byte. Amikor  $\lambda$  kisebb 75 kérés/másodperc-nél, a válaszidő Proxy Cache szerver használatával nagyobb mint Proxy használata nélkül. Azaz ebben az esetben igen magas  $\lambda > 75$  kell legyen a belső igények érkezési intenzitása, hogy megérje a Proxy Cache szerver üzemeltetése. A 3 grafikonon ugyanazokat a rendszer paramétereket használtuk, csak az  $a$  osztályhoz tartozó igények arányát növeltük meg 20%-ra. Mint ahogyan látható, ebben az esetben a válaszidők már  $\lambda > 65$  igény/másodperc esetén alacsonyabbak Proxy Cache szerver használatával. A 4 grafikonon azt az esetet látjuk, amikor az  $a$  osztályú fájlok esetében a találati valószínűséget növeljük  $p_a = 0.4$ -re. A grafikonon használt többi paraméter értékei: az  $a$  osztály aránya = 20%, a külső érkezési intenzitás  $\Lambda = 100$  igény/másodperc, a használt fájl méretek  $F_a = 7000$  byte,  $F_b = 1000$  byte, valamint a belső igények esetén a találati valószínűség  $p_a = 0.4$  és  $p_b = 0.25$ . A grafikonon látható, hogy ilyen magas találati valószínűség esetén a válaszidők minden belső igény érkezési intenzitás mellett alacsonyabbak Proxy Cache szerver használatával. Amennyiben csak az  $a$  osztályhoz tartozó fájlok találati valószínűségét csökkentjük, természetesen a Proxy Cache szerver hatékonysága drasztikusan romlik. Ezt láthatjuk a 5 grafikonon ahol a használt paraméterek megegyeznek a 4 grafikonon használt értékekkel, kivéve az  $a$  osztályú fájlok találati valószínűségét, ami  $p_a = 0.15$ . A grafikonok elemzésével látható, hogy a Proxy Cache szerver hatékonysága alacsonyabb találati valószínűség esetén, csak magas belső érkezési intenzitás mellett jár alacsonyabb válaszidőkkel. Viszont extrém magas találati valószínűség használatával ( $p_a = 0.4$ ) a Proxy Cache szerver használata minden esetben alacsonyabb válaszidőket eredményez.



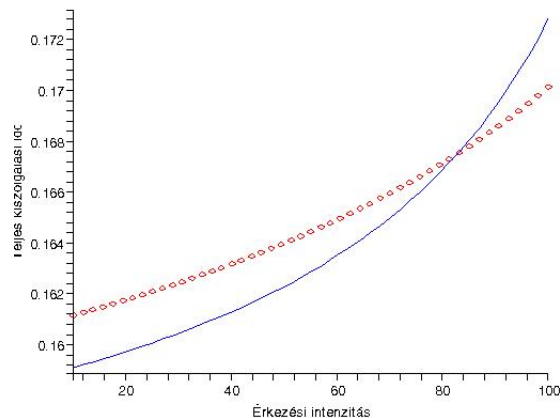
2. ábra. 10%  $a$  osztály,  $\Lambda = 100$ ,  $p_a = p_b = 0.25$ ,  $F_a = 7000$  bytes,  $F_b = 1000$  bytes



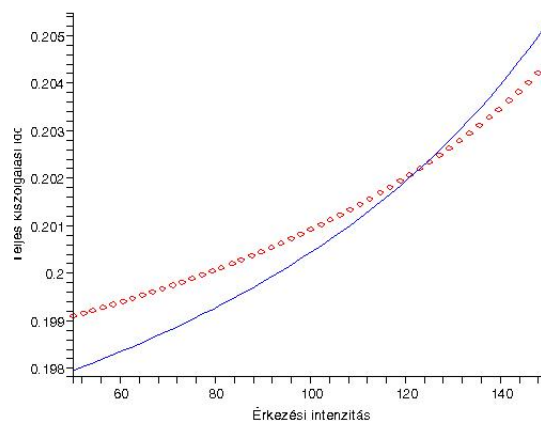
3. ábra. 20%  $a$  osztály,  $\Lambda = 100$ ,  $p_a = p_b = 0.25$ ,  $F_a = 7000$  bytes,  $F_b = 1000$  bytes



4. ábra. 20%  $a$  osztály,  $\Lambda = 100$ ,  $p_a = 0.4$ ,  $p_b = 0.25$ ,  $F_a = 7000$  bytes,  $F_b = 1000$  bytes



5. ábra. 20%  $a$  osztály,  $\Lambda = 100$ ,  $p_a = 0.15$ ,  $p_b = 0.25$ ,  $F_a = 7000$  bytes,  $F_b = 1000$  bytes



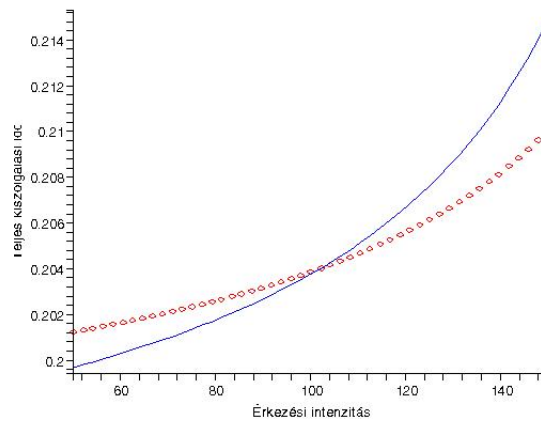
6. ábra. 30%  $a$  osztály,  $\lambda = 10$ ,  $p_a = p_b = 0.25$ ,  $F_a = 7000$  bytes,  $F_b = 1000$  bytes

### 3.2. A külső igények érkezési intenzitásának hatása a válaszidőre

A következő grafikonok segítségével a külső igények hatását fogjuk megvizsgálni. A 6 grafikonon használt paraméterek: az  $a$  osztály aránya = 30%, a belső igények érkezési intenzitása  $\lambda = 10$  igény/másodperc, a használt fájl méretek  $F_a = 7000$  byte,  $F_b = 1000$  byte, valamint a Proxy Cache szerveren a találati valószínűségek rendre  $p_a = p_b = 0.25$ . Amint látható, ha a külső igények érkezési intenzitása  $\Lambda > 125$  igény/másodperc alacsony belső érkezési intenzitás ( $\lambda = 10$ ) és viszonylag alacsony találati valószínűség ( $p_a = p_b = 0.25$ ) mellett is alacsonyabb válaszidőket kapunk Proxy Cache szerver használatával. A 7 grafikonon a használt paraméterek megegyeznek a 6 grafikon paramétereivel, csak a belső igények érkezési intenzitását növeltük  $\lambda = 50$ -re. Amint várható volt ebben az esetben már alacsonyabb külső érkezési intenzitás mellett is alacsonyabb válaszidőket kapunk Proxy Cache szerver használatával ( $\Lambda > 105$ ). Megvizsgálva a 6 - 7 grafikonokat általánosságban elmondhatjuk, hogy a külső igények érkezési intenzitásának növelésével a válaszidők nőnek függetlenül a Proxy Cache szerver jelenlététől. Amennyiben a külső igények intenzitása elég nagy, viszonylag kis belső érkezési intenzitás és találati valószínűség esetén is alacsonyabb válaszidőket kaphatunk Proxy Cache szerver használatával.

### 3.3. A fájl méret hatása a válaszidőre

A 8-9 grafikonokon a teljes válaszidőt az  $a$  osztályhoz tartozó fájl méretének függvényeként, míg a 10 grafikonon a  $b$  osztályhoz tartozó fájl méretének függvényeként ábrázoljuk. A 8 grafikonon a használt paraméterek értékei:



7. ábra. 30%  $a$  osztály,  $\lambda = 50, p_a = p_b = 0.25, F_a = 7000$  bytes,  $F_b = 1000$  bytes

Fájl méret( $a$ osztály)	Txc	T	Eltérés
$F_a = 2000$	0.09446809706	0.09317433644	0.00129376062
$F_a = 4000$	0.1217843586	0.1207717535	0.0010126051
$F_a = 6000$	0.1491608228	0.1484324701	0.0007283527
$F_a = 8000$	0.1766073005	0.1761686716	0.0004386289
$F_a = 10000$	0.2041361569	0.2039958857	0.0001402712
$F_a = 12000$	0.2317632395	0.2319342123	-0.0001709728
$F_a = 14000$	0.2595092442	0.2600101405	-0.0005008963
$F_a = 16000$	0.2874017905	0.2882593062	-0.0008575157
$F_a = 18000$	0.3154786670	0.3167308131	-0.0012521461

1. táblázat. Fájl méret hatása a válaszidőre, az  $a$  osztály aránya 20%

az  $a$  osztály aránya = 40%, a belső igények érkezési intenzitása  $\lambda = 50$  igény/másodperc, a külső igények érkezési intenzitása  $\Lambda = 100$  igény/másodperc, a használt  $b$  osztályhoz tartozó fájlok mérete  $F_b = 1000$  byte, valamint a Proxy Cache szerveren a találati valószínűségek rendre  $p_a = p_b = 0.25$ . Amint a grafikonon látható, az  $a$  osztályú fájl méret növelésével a válaszidők mind Proxy Cache szerver használatával, mind nélküle növekednek. Az ábrázolt két görbe csak  $F_a > 15000$  byte esetén távolodik el egymástól. A részletesebb vizsgálat érdekében a kapott pontos válaszidőket a 1 illetve a 2 táblázatokban láthatjuk, ahol az  $a$  osztály aránya rendre 20 illetve 40 százalék. A 1 táblázatban megfigyelhetjük, hogy kisebb  $a$  osztályú fájl méret esetén a Proxy Cache szerver használata nagyobb válaszidőket eredményez. De amint a fájl mérete eléri a 12000 byte-ot a válaszidők alacsonyabbak lesznek Proxy Cache szerver használatával. A 2 táblázatban a teljes válaszidőket láthatjuk amikor az  $a$  osztály aránya 40%. Megfigyelhetjük, hogy magasabb  $a$  osztály arány mellett a válaszidők szintén magasabbak, viszont a Proxy Cache szerver használatának az előnye már kisebb fájl méretnél megmutatkozik ( $F_a = 6000$  byte).

A 9 grafikonon az alap paraméterek változatlanok, egyedül az  $a$  osztály arányát növeltük meg 70%-ra. Megfigyelhetjük, hogy a grafikonon szereplő két görbe közötti eltérés számottevően nő a fájl méret növelésével, azaz magas  $a$  osztályú arány és nagy fájl méret használatával a Proxy Cache szerver használata kifizetődő. A 10 grafikonon a teljes válaszidőt a  $b$  osztályhoz tartozó fájl méretének függvényeként ábrázoljuk. A használt paraméterek: az  $a$  osztály aránya = 40%, a belső igények érkezési intenzitása  $\lambda = 50$  igény/másodperc, a külső igények érkezési intenzitása  $\Lambda = 100$  igény/másodperc, a használt  $a$  osztályú fájl mérete  $F_a = 7000$  byte valamint a Proxy Cache szerveren a találati valószínűségek rendre  $p_a = 0.25$  illetve  $p_b = 0.35$ .

Amint látható, Proxy Cache szerver használatával a használt paraméterek mellett a válaszidők végig kisebbek mint Proxy Cache szerver használata nélkül. Megfigyelhetjük, hogy a "normál" - azaz  $b$  osztályú fájl mérete 1000-2000 byte-os intervallumban lényegesen nem befolyásolja a  $T_{xc} - T$  különbséget.

A 3 táblázatban az  $a$  osztályhoz tartozó fájlok arányának hatását láthatjuk a válaszidőre. A használt pa-

Fájl méret( $a$ osztály)	T <sub>xc</sub>	T	Eltérés
$F_a = 2000$	0.1077452991	0.1067106059	0.0010346932
$F_a = 4000$	0.1624380248	0.1619687395	0.0004692853
$F_a = 6000$	0.2174133590	0.2175321551	-0.0001187961
$F_a = 8000$	0.2727864463	0.2735464100	-0.0007599637
$F_a = 10000$	0.3287558693	0.3302670825	-0.0015112132
$F_a = 12000$	0.3856999610	0.3881879423	-0.0024879813
$F_a = 14000$	0.4444483664	0.4484027240	-0.0039543576
$F_a = 16000$	0.5072639578	0.5139139582	-0.0066500004
$F_a = 18000$	0.5833834923	0.5970200201	-0.0136365278

2. táblázat. Fájl méret hatása a válaszidőre, az  $a$  osztály aránya 40%

Az $a$ osztály aránya	T <sub>xc</sub>	T	Eltérés
10	0.1219838832	0.1209124357	0.0010714475
20	0.1628746062	0.1622902553	0.0005843509
30	0.2038840246	0.2037976778	0.0000863468
40	0.2450404415	0.2454704817	-0.0004300402
50	0.2863831749	0.2873589491	-0.0009757742
60	0.3279687079	0.3295360446	-0.0015673367
70	0.3698815197	0.3721118312	-0.0022303115
80	0.4122543652	0.4152604812	-0.0030061160
90	0.4553092755	0.4592749103	-0.0039656348

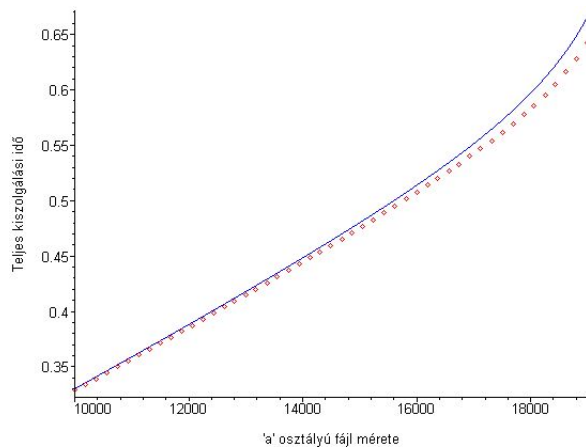
3. táblázat. Az  $a$  osztály arányának hatása a válaszidőre

raméterek: a belső igények érkezési intenzitása  $\lambda = 50$  igény/másodperc, a külső igények érkezési intenzitása  $\Lambda = 100$  igény/másodperc, a használt  $a$  illetve  $b$  osztályú fájl méretek  $F_a = 7000$  byte illetve  $F_b = 1000$  byte valamint a Proxy Cache szerveren a találati valószínűségek rendre  $p_a = 0.25$  illetve  $p_b = 0.25$ . Láthatjuk, hogy az  $a$  osztályhoz tartozó tartalom arányának növelésével a válaszidők nőnek függetlenül attól, hogy installáltunk-e Proxy Cache szervert vagy sem. Valamint megfigyelhetjük, hogy az  $a$  osztályhoz tartozó fájlok arányának növelésével a különbség a két válaszidő között ( $T_{xc} - T$ ) egyre kisebb és 40% fölötti  $a$  osztályú tartalom esetén a használt paraméter értékek mellett, Proxy Cache szerver használatával már alacsonyabb válaszidőket kapunk, mint Proxy Cache szerver használata nélkül.

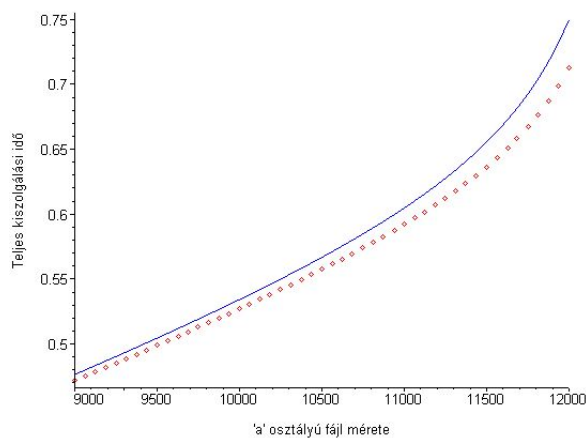
## 4. Összefoglaló

Módosítottuk a [2] cikkben szereplő Proxy Cache szerver modellt, hogy vizsgálni lehessen a heterogén forgalom hatását a Proxy Cache szerverek hatékonyságára. A kliensek által keresett fájlokat méretük alapján két osztályba soroltuk. Az átlagosnál nagyobb méretű fájlok az  $a$ , míg a kis méretű, úgynevezett "normál" fájlok a  $b$  osztályba kerülnek. Mindkét osztályba tartozó igény esetén először megvizsgáljuk, hogy a fájl megtalálható-e a Proxy Cache szerveren vagy sem. Ezt a találati valószínűséget  $p_a$  illetve  $p_b$  -vel jelöljük az  $a$  illetve  $b$  osztályba tartozó fájlok esetén. Amennyiben a keresett fájl megtalálható a Proxy Cache szerveren, akkor mindkét osztály esetén a fájl egy másolata azonnal továbbítódik a klienshez. Ellenkező esetben, amikor is a fájl nem található meg a Proxy Cache szerveren az igény továbbítódik a távoli Web szerverhez függetlenül az osztályától. Miután az igényelt fájl visszaérkezik a Proxy Cache szerverhez egy másolat továbbítódik a klienshez.

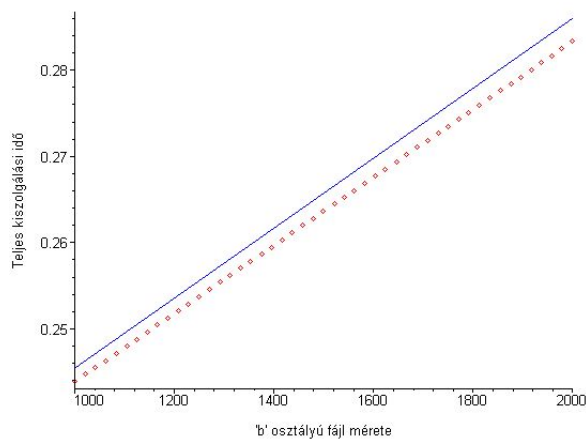
Feltételeztük, hogy mindkét osztályhoz tartozó igények a Proxy Cache szerverhez Poisson-folyamat szerint érkeznek, és a Web szerverhez kívülről érkező igények szintén Poisson-folyamat alapján érkeznek, valamint mind a Proxy Cache szerver mind pedig a Web szerver kiszolgálási ideje független exponenciális eloszlású valószínűségi változó. Kiszámítottuk egy tetszőleges belső igény válaszidejét Proxy Cache szerver használata esetén, valamint Proxy Cache szerver használata nélkül. Így vizsgálni lehetett a Proxy Cache szerver használatát különböző paraméterértékek mellett.



8. ábra. 40%  $a$  osztály,  $\lambda = 50$ ,  $\Lambda = 100$ ,  $p_a = 0.25$ ,  $p_b = 0.25$ ,  $F_b = 1000$  bytes



9. ábra. 70%  $a$  osztály,  $\lambda = 50$ ,  $\Lambda = 100$ ,  $p_a = 0.25$ ,  $p_b = 0.25$ ,  $F_b = 1000$  bytes



10. ábra. 40%  $a$  osztály,  $\lambda = 50$ ,  $\Lambda = 100$ ,  $p_a = 0.25$ ,  $p_b = 0.35$ ,  $F_a = 7000$  bytes

4. táblázat. **Heterogén forgalmi modell paramétere**

$\lambda_a$ :	belső $a$ osztályú igények érkezési intenzitása
$\lambda_b$ :	belső $b$ osztályú igények érkezési intenzitása
$\Lambda_a$ :	külső $a$ osztályú igények érkezési intenzitása
$\Lambda_b$ :	külső $b$ osztályú igények érkezési intenzitása
$F_a$ :	az $a$ osztályhoz tartozó fájl mérete (byte-ban)
$F_b$ :	a $b$ osztályhoz tartozó fájl mérete (byte-ban)
$p_a$ :	találati valószínűség az $a$ osztályhoz tartozó fájlok esetén
$p_b$ :	találati valószínűség a $b$ osztályhoz tartozó fájlok esetén
$B_{xc}$ :	a Proxy cache szerver kimenő puffere (byte-ban)
$I_{xc}$ :	a Proxy Cache szerver keresési ideje (másodpercben)
$Y_{xc}$ :	a PCS Statikus szerver ideje (másodpercben)
$R_{xc}$ :	a dinamikus szerver arány a Proxy Cache szerveren (byte/másodperc)
$N_c$ :	kliens hálózati sávszélesség (bit/másodperc)
$B_s$ :	Web szerver kimenő puffere (byte-ban)
$I_s$ :	Inicializálási idő (másodpercben)
$Y_s$ :	a Web szerver statikus szerver ideje (másodperc)
$R_s$ :	a Web szerver dinamikus szerver aránya (byte/másodperc)
$N_s$ :	kliens hálózati sávszélesség (bit/másodperc)

Megvizsgáltuk a Proxy Cache szerver hatékonyságát a belső valamint külső érkezési intenzitások függvényében, valamint vizsgáltuk az  $a$  illetve  $b$  osztályhoz tartozó fájlok méretének és az  $a$  illetve  $b$  osztály arányának hatását a válaszidőre. Megmutattuk, hogy mind a belső mind pedig a külső érkezési intenzitás növelésével a válaszidők nőnek, függetlenül a Proxy Cache szerver jelenlététől. Amennyiben az  $a$  osztályos kérések arányát növeljük a válaszidők szintén nőnek, valamint magas  $a$  osztály arányt használva már alacsonyabb érkezési intenzitás esetén is megéri a Proxy Cache szerver használata. Alacsony  $a$  osztály arány, alacsony érkezési intenzitás és alacsony találati valószínűség használatával Proxy Cache szerverrel magasabb válaszidőket kapunk mint nélküle.

## Hivatkozások

- [1] AGGARWAL, C., WOLF, J.L. and YU, P.S. Caching on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, **11** (1999), 94-107.
- [2] BERCZES, T., GUTA, G., KUSPER, G., SCHREINER, W. and SZTRIK, J., Analyzing Web Server Performance Models with the Probabilistic Model Checker PRISM. *Technical report no. 08-17 in RISC Report Series*,
- [3] BOSE, I., CHENG, H.K., Performance models of a firms proxy cache server. *Decision Support Systems and Electronic Commerce.*, **29** (2000), 45-57.
- [4] CACHEFLOW INC., 1999. CacheFlow White Papers. Available from <http://cacheflow.com/technology/wp/>.
- [5] CAUGHEY, S.J., INGHAM, D.B. and LITTLE, M.C. Flexible open caching for the Web. *Computer Networks and ISDN Systems*, **29** (1997), 1007-1017.
- [6] LASHINSKY, A., Suddenly cache is king the world of net stocks. *Fortune.*, (1999), 370-372.
- [7] MENASCE, D.A., ALMEIDA, V.A.F., Capacity Planning for Web Performance: Metric, Models, and Methods. *Prentice Hall.*, (1998)
- [8] RUBENSTEIN, R., HERSCH, H.M., LEDGARD, H.F., The Human Factor: Designing Computer Systems for People. *Digital Press, Burlington, MA.*, (1984)
- [9] ZHAO, J.L., KUMAR, A., Data management issues for large scale, distributed workflow systems on the internet. *ACM SIGMIS Data Base.*, **29** (4), 22-32.
- [10] L.P. SLOTHOUBER, A model of Web server performance. *5th International World Wide Web Conference, Paris, France.*, (1996)