Symulacja w Badaniach i Rozwoju
Vol. 5, No. 3/2014

Walenty ONISZCZUK
Politechnika Białostocka, Wydział Informatyki
ul. Wiejska 45A , 15-351 Białystok
E-mail: w.oniszczuk@pb.edu.pl

Zjawiska dławień intensywności zadań do serwerów – eksperymenty symulacyjne

1 Wstęp

Procesy dławienia (zmniejszania) intensywności napływu nowych zadań ze źródeł, o wyższym oraz niższym priorytecie, do bloku serwerów (stanowisk obsługi) są ważnymi działaniami, związanymi z zapobieganiem nadmiernym przeciążeniom tychże serwerów. Dotyczą one szeroko rozumianych procedur dostępu użytkowników do zasobów systemów i sieci komputerowych. Takie procedury regulujące dostęp użytkowników do węzła obsługi (bloku serwerów) z reguły łączone są ze specjalnymi, dynamicznymi procedurami manipulowania progami we wspólnych buforach [2, 3].

Pamiętając o tym, że pojemność buforów w węzłach obsługi jest zawsze ograniczona, przyjmuje się specjalne polityki (procedury) manipulowania ich wielkościami lub stosuje się procedury dławienia intensywności napływu nowych zadań, które można zrealizować przez wprowadzenie czasowych blokad ich źródeł. Można też stosować obie strategie naraz. Dokładniej, wspomniane powyżej procedury blokad rozumiemy i realizujemy jako czasowe wstrzymania transferów zadań ze źródeł do serwerów, a potem ich automatyczne wznawianie, w zależności od zapełnienia buforu zadaniami wyższego i niższego priorytetu [1, 5].

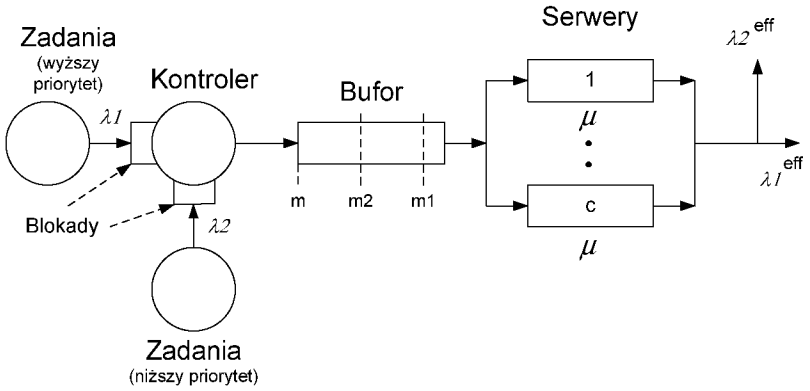
Łącząc algorytmy czasowego wstrzymywania transferów zadań ze źródeł (blokady) z polityką/algorytmami przesuwania (manipulowania) progami w buforach, otrzymamy blok zarządzania, traktowany jako pewien stochastyczny automat ze zmienną strukturą, działający w typowo losowym środowisku napływu i obsługi zadań. Mechanizmy tego typu mogą być ulokowane na przykład w modułach o nazwie Kontroler, na wejściu do węzłów obsługi z buforami (patrz rys. 1).

Aby badać takie stochastyczne automaty, regulujące procesy napływu i obsługi zadań, należy zdefiniować i opisać analityczny model pracy takich układów, a następnie utworzyć pakiet programowy (w wybranym języku programowania), który pozwoli przeprowadzić serię eksperymentów symulacyjnych, obrazujących dynamikę takich systemów komputerowych. Eksperymenty te pokażą, na ile ten czy inny czynnik, jak wielkość bufora, usytuowanie progów, zmiany intensywności napływu zadań wyższego czy niższego priorytetu, zmiany intensywności obsługi (mocy serwerów), wpływają na wskaźniki wydajności i jakości obsługi węzłów obsługi [6].

Uproszczony model takiego układu: węzły generujące zadania plus stanowiska obsługi z buforem i progami pokazany jest na rysunku numer 1. W tak zdefiniowanej konfiguracji sieci zadania wyższego i niższego priorytetu, poprzez moduł Kontrolera, przechodzą od źródeł do bufora, który jest wspólny dla wszystkich serwerów węzła obsługi. Moduł Kontrolera, poprzez algorytm częściowego podziału bufora progami,

kontroluje intensywność (ruch) napływających zadań, blokując jedno lub oba źródła i zatrzymując transfer zadań, a potem odpowiednio wznawiając transmisje.

Jeżeli zapełnienie bufora zadaniami jest mniejsze niż próg wejściowy *m2*, to Kontroler akceptuje zadania obu priorytetów (klas), a jeżeli przekroczy ten próg, to zadania niższego priorytetu są blokowane w źródle. Wznawianie ich transmisji nastąpi dopiero po przekroczeniu progu powrotnego *m1* (zjawisko histerezy).



Rys.1. Model węzła serwerów z blokadami i progami
Fig.1. Model of servers' node with blocking and thresholds

Zadania wyższego priorytetu są przyjmowane zawsze, niezależnie od ustawienia (polityki) progów, a blokada ich źródła nastąpi dopiero po pełnym zapełnieniu bufora, zaś wznowienie transmisji po pojawieniu się wolnego miejsca w buforze. Zjawiska blokad źródeł, przerywania i wznawiania transmisji można traktować jako klasyczny mechanizm kontroli i dławienia intensywności napływu nowych zadań do bloku serwerów. Naczelną zasadą jest tutaj racjonalny dopływ zadań do węzła obsługi. Aby to zrealizować, tworzymy adaptacyjny algorytm manipulacji progami, połączony ze zjawiskiem blokad. Pozwala to w sposób bardziej racjonalny gospodarować i dzielić pamięć buforów w zależności od aktualnej sytuacji panującej w sieci, a polityka dynamicznych progów ułatwia dostosowanie się do aktualnej intensywności napływających zadań.

2 Model matematyczny

Analiza pracy układu źródła zadań – serwery, pokazanego na rysunku 1, i zbudowanie jego analitycznego modelu oparta na następujących pryncypialnych założeniach:

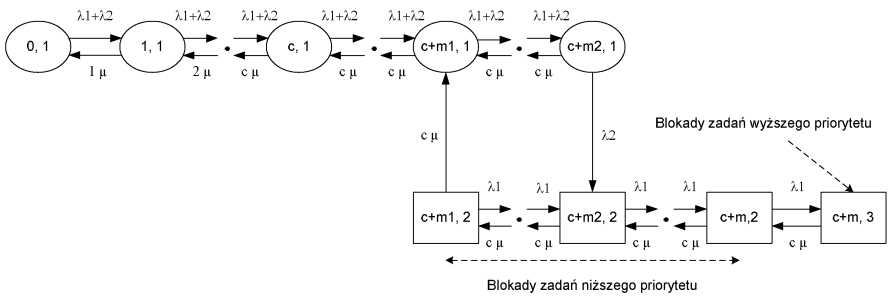
1. Węzeł obsługi składa się z c równoległych linii (serwerów), ze wspólnym buforem.
2. Czas obsługi zadań w węźle – zmienna losowa o rozkładzie wykładniczym, z intensywnością μ.
3. Pojemność bufora równa jest m, z dynamicznymi progami *m1* i *m2*.
4. Zdefiniowane są dwa bloki (węzły) generujące zadania wyższego i niższego priorytetu.

5. Procesy napływu zadań: procesy Poissona z intensywnościami $\lambda 1$ i $\lambda 2$.
6. Blok kontrolera jest blokiem decyzyjnym, usytuowanym przed wejściem do bufora.

Przy tak zdefiniowanych i przyjętych podstawowych założeniach układ źródła zadań – serwery można zdefiniować jako model kolejkowy, który z kolei może być reprezentowany przez łańcuch Markowa, z ciągłym czasem i dyskretną przestrzenią stanów. Taki proces Markowa można badać dla stacjonarnych/ustabilizowanych warunków, gdzie łańcuch Markowa ma stacjonarne prawdopodobieństwa wszystkich, fizycznie możliwych stanów. Zakładając, że bufor do serwerów ma ograniczoną pojemność, otrzymujemy łańcuch o ograniczonej przestrzeni stanów. Rozwiązując numerycznie łańcuch Markowa, można obliczyć stacjonarne prawdopodobieństwa stanów, a z nich miary wydajności i jakości obsługi układu.

Klasa modeli kolejkowych z dynamicznymi progami i blokadami dokładniej analizowana i opisana jest, w innej pracy autora (patrz [4]), a w tej pracy nacisk położony jest na eksperymenty symulujące działanie układu, dla całej serii zmieniających się danych wejściowych, odzwierciedlających dynamikę modelu traktowanego jako stochastyczny automat ze zmienną strukturą.

W tak zdefiniowanej markowowskiej sieci kolejkowej pierwszym i podstawowym etapem analizy jest określenie pełnego grafu stanów modelu, a w nim określenie wszystkich intensywności przejść między stanami (rys. 2). Każdy ze stanów prezentowanego modelu sieci może być opisany przez parę indeksów (*i, k*), gdzie *i* to liczba zadań użytkowników w węźle obsługi, a *k* to stan serwerów węzła. Indeksy *k* to: *0* - przestój serwerów, *1* - obsługa zadań obu klas (priorytetów), *2* – obsługa zadań wyższego priorytetu i blokada źródeł o niższym priorytecie (przerwanie transferu), *3* – blokada źródeł zadań wyższego priorytetu.



Rys. 2. Diagram stanów sieci

Fig. 2. Network state transition diagram

Z grafu stanów można od razu zestawić układy równań algebraicznych do obliczania stacjonarnych prawdopodobieństw wszystkich wyróżnionych stanów modelu. I tak dla stanów bez blokad mamy:

$(\lambda_1+\lambda_2)\cdot p_{0,1} = \mu\cdot p_{1,1}$;

$(\lambda_1+\lambda_2+i\cdot\mu)\cdot p_{i,1} = (\lambda_1+\lambda_2)\cdot p_{i-1,1} + (i+1)\cdot\mu\cdot p_{i+1,1}$ dla $i = 1, \ldots, c-1$;

$(\lambda_1+\lambda_2+c\cdot\mu)\cdot p_{i,1} = (\lambda_1+\lambda_2)\cdot p_{i-1,1} + c\cdot\mu\cdot p_{i+1,1}$ dla $i = c, \ldots, c+m1-1$;

$(\lambda_1+\lambda_2+c\cdot\mu)\cdot p_{c+m1,1} = (\lambda_1+\lambda_2)\cdot p_{c+m1-1,1} + c\cdot\mu\cdot p_{c+m1+1,1} + c\cdot\mu\cdot p_{c+m1,2}$;

$(\lambda_1+\lambda_2+c\cdot\mu)\cdot p_{i,1} = (\lambda_1+\lambda_2)\cdot p_{i-1,1} + c\cdot\mu\cdot p_{i+1,1}$ dla $i = c+m1+1, \ldots, c+m2-1$;

$(\lambda_2+c\cdot\mu)\cdot p_{c+m2,1} = (\lambda_1+\lambda_2)\cdot p_{c+m2-1,1}$.

Z kolei dla stanów z blokadami mamy:

$(\lambda_1+c\cdot\mu)\cdot p_{c+m1,2} = c\cdot\mu\cdot p_{c+m1+1,2}$;

$(\lambda_1+c\cdot\mu)\cdot p_{i,2} = \lambda_1\cdot p_{i-1,2} + c\cdot\mu\cdot p_{i+1,2}$ dla $i = c+m1+1, \ldots, c+m2-1$;

$(\lambda_1+c\cdot\mu)\cdot p_{c+m2,2} = \lambda_1\cdot p_{c+m2-1,2} + \lambda_2\cdot p_{c+m2,1} + c\cdot\mu\cdot p_{c+m2+1,2}$;

$(\lambda_1+c\cdot\mu)\cdot p_{i,2} = \lambda_1\cdot p_{i-1,2} + c\cdot\mu\cdot p_{i+1,2}$ dla $i = c+m2+1, \ldots, c+m-1$;

$(\lambda_1+c\cdot\mu)\cdot p_{c+m,2} = \lambda_1\cdot p_{c+m-1,2} + c\cdot\mu\cdot p_{c+m,3}$;

$c\cdot\mu\cdot p_{c+m,3} = \lambda_1\cdot p_{c+m,2}$.

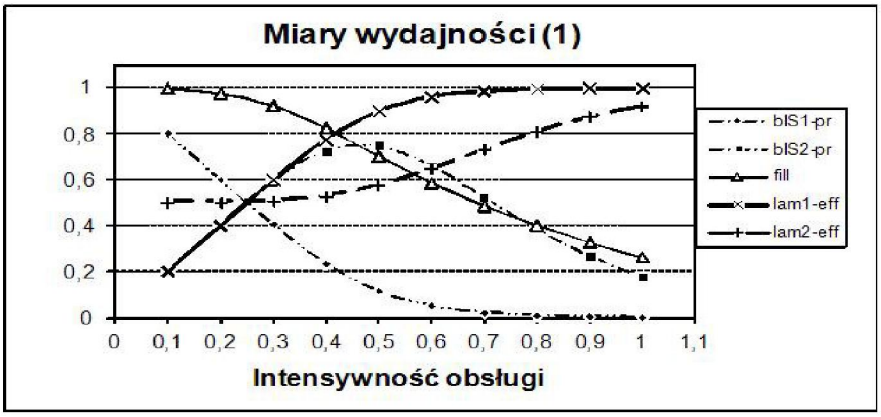
Takie układy równań rozwiązywane są metodami numerycznymi w standardowy sposób. Procedura ich rozwiązania oraz algorytmy liczenia podstawowych miar wydajności i jakości obsługi umieszczone są w specjalnie do tego celu stworzonym pakiecie programowym. Pakiet ten wykorzystano do przeprowadzenia serii eksperymentów symulacyjnych, badających przede wszystkim zjawiska dławienia (ograniczania) intensywności napływu zadań do węzła obsługi poprzez algorytmy podziału pojemności bufora dla zadań obu priorytetowych klas, a przede wszystkim poprzez wprowadzenie i zdefiniowanie procedur blokady źródeł w zależności od aktualnie istniejącej sytuacji w sieci (przeciążenia).

3 Eksperymenty symulacyjne

Do demonstracji dynamiki pracy węzłów obsługi z kilkoma serwerami obsługującymi użytkowników dwóch priorytetowych klas, z blokadami i dynamicznymi procedurami manipulacji pojemnościami buforów, przeprowadzono całą serię eksperymentów symulacyjnych. Wykorzystano do tego pakiet programowy stworzony zgodnie z procedurami analizy z sekcji 2 oraz z innymi procedurami obliczeń, dokładniej pokazanymi w innej publikacji autora (patrz [4]).

Seria 1. Badanie wpływu zmian intensywności obsługi w węźle na miary wydajności i jakości obsługi (**OoS**) sieci. Dla arbitralnie dobranej konfiguracji układu źródła zadań – serwery węzła obsługi przyjęto następujące parametry wejściowe: $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\boldsymbol{\mu = var\ [0.1 - 1.0]}$, $c = 2$, $m = 10$, $m1 = 3$, $m2 = 6$.

Częściowe wyniki eksperymentów pokazano na rysunku 3.



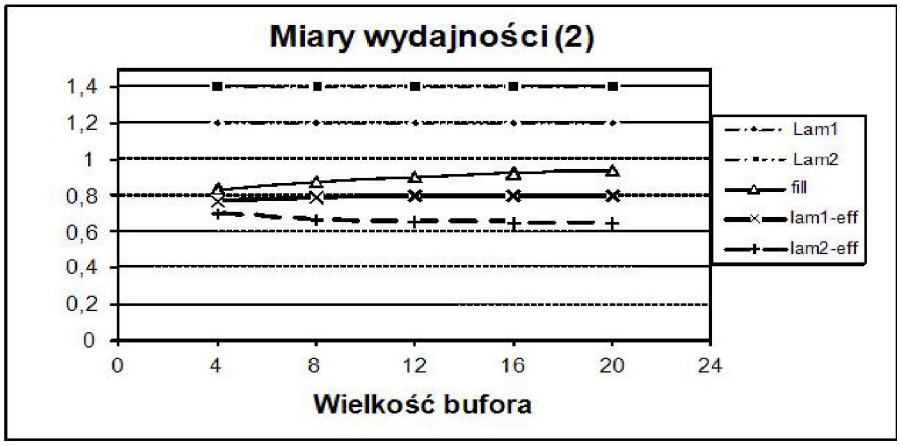
Rys. 3. Diagram parametrów QoS, gdzie: ***blS1-pr*** *to prawdopodobieństwo blokady źródeł o wyższym priorytecie,* ***blS2-pr*** *to prawdopodobieństwo blokady źródeł o niższym priorytecie,* ***fill*** *to parametr zapełnienia bufora,* ***lam1-eff*** *oraz* ***lam2-eff*** *to realne (efektywne) intensywności strumieni wyższego i niższego priorytetu*

Fig. 3. Graphs of QoS parameters, where: ***blS1-pr*** *is the high priority source blocking probability,* ***blS2-pr*** *is the low priority source blocking probability,* ***fill*** *is the buffer filling parameter,* ***lam1-eff*** *and* ***lam2-eff*** *are the effective arrival rates of high and low priority jobs, respectively*

Komentarze. Przy nominalnej (deklarowanej) intensywności obu klas strumieni zadań równej *1.0* na umowną jednostkę czasu realna intensywność zadań wyższego priorytetu zmienia się *0.2* (pięciokrotne dławienie), by w drugiej połowie wykresu zbliżyć się do intensywności nominalnej. Z kolei intensywność zadań niższego priorytetu zwiększa się dużo wolniej, od początkowej *0.5* do *0.9* na końcu wykresu. W obu przypadkach szybkość obsługi serwerów równomiernie zwiększa się od *0.1* do *1.0* na umowną jednostkę czasu, co skutkuje zmniejszaniem się prawdopodobieństw obu blokad (wykresy *blS1-pr* i *blS2-pr*), a to z kolei powoduje zmniejszanie się współczynników dławienia strumieni zadań.

Seria 2. Wpływ zmian wielkości bufora i usytuowania progów na miary wydajności i jakości obsługi w sieci. Dla arbitralnie dobranej konfiguracji układu źródła zadań – serwery węzła obsługi, przyjęto następujące parametry wejściowe: $\lambda_1 = 1.2$, $\lambda_2 = 1.4$, $\mu = 0.2$, $c = 4$, $\mathbf{m = var\ [4 - 20]}$, $\mathbf{m1 = m/4, m2 = m/2}$.

Częściowe wyniki eksperymentów pokazano na rysunku 4.



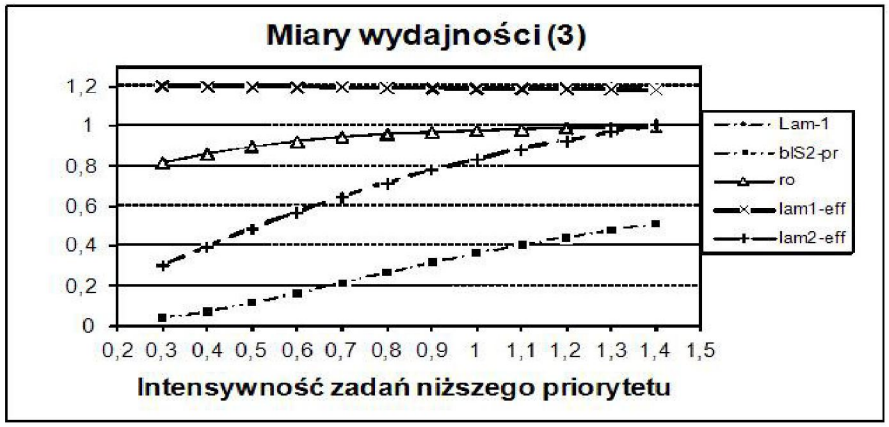
Rys. 4. Diagram parametrów QoS, gdzie: ***Lam1*** *oraz* ***Lam2*** *to deklarowane intensywności zadań wyższego i niższego priorytetu,* ***fill*** *to parametr zapełnienia bufora,* ***lam1-eff*** *oraz* ***lam2-eff*** *to odpowiednio realne (efektywne) intensywności strumieni wyższego i niższego priorytetu*

Fig. 4. Graphs of QoS parameters, where: ***Lam1*** *and* ***Lam2*** *are the nominal arrival rates of high and low priority jobs, respectively,* ***fill*** *is the buffer filling parameter,* ***lam1-eff*** *and* ***lam2-eff*** *are the effective arrival rates of high and low priority jobs, respectively*

Komentarze. Zaskakująco nieduże zmiany współczynników dławienia strumieni, przy dość znacznych zmianach pojemności bufora (pięciokrotność) i miejsca usytuowania progów. Przy nominalnej intensywności zadań wyższego priorytetu równej *1.2* efektywna intensywność wzrasta od *0.77* do *0.80*, zaś dla zadań niższego priorytetu spada od *0.70* do *0.65*, przy nominalnej intensywności równej *1.4*. Reasumując, na dławienie wpływ ma przede wszystkim moc serwerów i ich liczba.

Seria 3. Badanie wpływu zmian intensywności strumienia zadań niższego priorytetu na miary wydajności i jakości obsługi w sieci. Dla arbitralnie dobranej konfiguracji układu źródła zadań – serwery węzła obsługi przyjęto następujące parametry wejściowe: $\lambda_1 = 1.2$, $\boldsymbol{\lambda_2 = var\ [0.3 - 1.4]}$, $\mu = 0.6$, $c = 3$, $m = 12$, $m1 = 4$, $m2 = 10$.

Częściowe wyniki tych eksperymentów pokazano na rysunku 5.



Rys. 5. Diagram parametrów QoS, gdzie: ***Lam1*** *to nominalna intensywność napływu zadań wyższego priorytetu,* ***blS2-pr*** *to prawdopodobieństwo blokady źródeł o niższym priorytecie,* ***ro*** *to współczynnik obciążenia węzła,* ***lam1-eff*** *oraz* ***lam2-eff*** *to realne (efektywne) intensywności strumieni wyższego i niższego priorytetu*

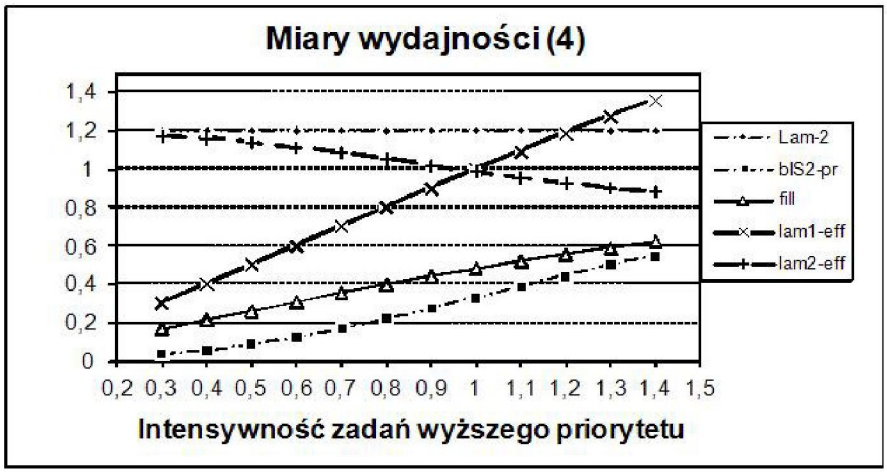
Fig. 5. Graphs of QoS parameters, where: ***Lam1*** *is the nominal arrival rate of high priority jobs,* ***blS2-pr*** *is the low priority source blocking probability,* ***ro*** *is the buffer utility parameter,* ***lam1-eff*** *and* ***lam2-eff*** *are the effective arrival rates of high and low priority jobs, respectively*

Komentarze. Przy wielokrotnej zmianie nominalnej intensywności zadań niższego priorytetu, od *0.3* do *1.4*, zaobserwowano bardzo mały współczynnik dławienia intensywności zadań wyższego priorytetu, zaś realna (efektywna) intensywność zadań niższej klasy zmieniała się od *0.3* do *1.0*. Ważnym czynnikiem jest tutaj ustawienie progów i to determinuje dynamikę zmian układu.

Seria 4. Wpływ zmian intensywności strumienia zadań wyższego priorytetu na miary wydajności i jakości obsługi w sieci. Dla arbitralnie dobranej konfiguracji układu źródła zadań – serwery węzła obsługi przyjęto następujące parametry wejściowe: λ_1 = ***var [0.3 - 1.4]***, $\lambda_2 = 1.2$, $\mu = 0.6$, $c = 3$, $m = 12$, $m1 = 4$, $m2 = 10$.

Częściowe wyniki eksperymentów pokazano na rysunku numer 6.

Komentarze. W tej serii eksperymentów badany był wpływ fluktuacji (zmian) intensywności strumienia zadań wyższego priorytetu na współczynnik dławienia intensywności zadań niższego priorytetu. Wybrana konfiguracja sieci była identyczna jak w trzeciej serii eksperymentów. Eksperymenty pokazały, że efektywna (realna) intensywność zadań wyższego priorytetu ulega minimalnemu dławieniu: od nominalnej *0.3* do *1.36* (zamiast *1.4*), zaś intensywność zadań niższego priorytetu systematycznie spada od *1.17* do *0.88*, zamiast nominalnej *1.2*. Ilustruje to wybraną strategię elastycznego zarządzania buforami w zależności od zmieniającego się obciążenia węzła serwerów zadaniami obu priorytetowych klas.



Rys. 6. Diagram parametrów QoS, gdzie: ***Lam2*** *to nominalna intensywność napływu zadań niższego priorytetu,* ***blS2-pr*** *to prawdopodobieństwo blokady źródeł o niższym priorytecie,* ***fill*** *to parametr zapełnienia bufora,* ***lam1-eff*** *oraz* ***lam2-eff*** *to realne (efektywne) intensywności strumieni wyższego i niższego priorytetu*

Fig. 6. Graphs of QoS parameters, where: ***Lam2*** *is the nominal arrival rate of low priority jobs,* ***blS2-pr*** *is the low priority source blocking probability,* ***fill*** *is the buffer filling parameter,* ***lam1-eff*** *and* ***lam2-eff*** *are the effective arrival rates of high and low priority jobs, respectively*

4 Podsumowanie

Wyniki eksperymentów pokazują dynamikę pracy sieci składającej się z bloku serwerów i dwóch źródeł zadań. Źródła te generują ruch teleinformatyczny wyższego i niższego priorytetu, a badania pokazują, jak wielki jest wpływ zjawisk blokad (czasowego wstrzymywania transferu zadań do węzła obsługi) i polityki dynamicznej manipulacji progami w buforze na miary wydajności i jakości obsługi sieci. Te zjawiska bezpośrednio wpływają na realne intensywności transferu danych od użytkowników do węzła obsługi. Dotyczy to zadań tak pierwszej, jak i niższej klasy. Głównym celem i zadaniem bloku Kontrolera, na wejściu do węzła obsługi, jest regulowanie napływu nowych zadań w zależności do aktualnej sytuacji procesu ich obsługi (eliminacja przeciążeń przez blokady i zarządzanie progami w buforze).

Literatura

1. Kwiecień J., Filipowicz B.: Firefly algorithm in optimization of queueing systems, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 60, no. 2, pp. 363-368, 2012
2. Oniszczuk W.: An Intelligent Service Strategy in Linked Networks with Blocking and Feedback, *Studies in Computational Intelligence N. 134 "New Challenges in Applied Intelligence Technologies"*, N.T. Nguyen, R. Katarzyniak (eds.), Springer-Verlag, Berlin, Heidelberg, pp. 351-361, 2008

3. Oniszczuk W.: Semi-Markov-based approach for analysis of open tandem networks with blocking and truncation, *International Journal of Applied Mathematics and Computer Science*, vol. 19(1), pp. 151-163, 2009
4. Oniszczuk W.: Flexible buffer management with thresholds and blocking for congestion control in multi-server computer systems, *Theoretical and Applied Informatics*, vol. 20, no.1, pp. 49-61, 2010
5. Oniszczuk W.: Loss Tandem Networks with Blocking Analysis – A Semi-Markov Approach, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 58, no. 4, pp. 673-681, 2010
6. Perros H.G.: *Queuing Networks with Blocking. Exact and Approximate Solution*, Oxford University Press, New York 1994

Streszczenie

W pracy przedstawiono wyniki eksperymentów symulacyjnych ukierunkowanych na badania zjawisk dławień intensywności strumieni zadań napływających do bloku serwerów. Badana była konfiguracja sieci z blokadami i dynamiczną manipulacją progami w buforach. Te ograniczające mechanizmy zrealizowano poprzez specjalny moduł kontroli napływu nowych zadań (dwóch priorytetowych klas) do wspólnych węzłów obsługi. Taki moduł kontroli zawiera w sobie adaptacyjne algorytmy manipulowania progami w buforach, reagujących na bieżące zmiany ruchu teleinformatycznego w sieci. Wyniki eksperymentów pokazują, jak ważnym czynnikiem są mechanizmy blokad i dynamicznych progów w sieciach z ograniczonymi buforami.

Słowa kluczowe: alokacja zadań i zasobów, blokady, systemy z progami, elastyczne zarządzanie buforami

Simulation experiments with phenomena of stifling jobs intensities to computer servers

Summary

This paper presents the series of experiments with simulation of stifling job intensities in some computer systems/networks with flexible buffer management and blocking. These constrains are treated as some control schemes for two priority job classes models in congested computer systems. The proposed scheme incorporates adaptive thresholds, which dynamically adjust according to computer system traffic behavior changes. The results of experiments confirm importance of a special treatment for the models with blocking, and threshold policy, in finite capacity buffers, which justifies this research.

Keywords: task and resource allocation, blocking, threshold-base systems, flexible buffer management