

Streszczenie rozprawy doktorskiej „Router interface as a queuing system with correlated service times”

Krzysztof Rusek

4 maja 2016

1 Wstęp

Rozprawa dotyczy modelowania interfejsu routera w celu obliczania charakterystyk kolejki. Głównym zadaniem przedstawionej analizy jest zbadanie architektury pamięci urządzenia oraz jej implikacja w modelach analitycznych. Wiele z istniejących badań odnosi się do modelu samego ruchu. Poprawność tych modeli wraz z modelem interfejsu routera nie jest jednak tak często weryfikowana. Eksperymenty na rzeczywistym urządzeniu pokazują jednak, że nieprawidłowy model interfejsu może spowodować znaczne różnice między prawdopodobieństwem odrzucenia pakietu mierzonym na interfejsie fizycznym i pochodzącym z symulacji.

W rozprawie zaproponowano dokładny model interfejsu routera (router na bazie Cisco IOS o nie-rozproszonej architekturze), który zweryfikowano eksperymentalnie na fizycznym urządzeniu dowodząc następującą tezę: **Interfejs urządzenia sieciowego, którego bufor może pomieścić określoną liczbę pakietów bez względu na ich długość, można modelować jako system kolejkowy MAP/SM/1/b o skorelowanych czasach obsługi. Dla takiego modelu możliwe jest wyznaczenie wybranych charakterystyk kolejkowania pakietów w stanie przejściowym i ustalonym.**

Interfejs jest modelowany jako bufor ograniczony określoną liczbą pakietów (niezależnie od ich długości) a nie rozmiarem pamięci. Bufor ten jest opisany jako skończony system kolejkowy FIFO zasilany przez markowski proces zgłoszeń (MAP) z półmarkowowskim (SM) czasem obsługi ($MAP/SM/1/b$ w notacji Kendalla). Takie założenia pozwalają uzyskać nowe wyniki analityczne dla charakterystyk kolejkowania pakietów. Wyznaczone zostały następujące charakterystyki: czas do przepełnienia bufora, lokalne natężenia

strat, współczynnik strat oraz łączna liczba strat w danym przedziale czasowym. Przewidywania proponowanego modelu mogą być nawet kilka rzędów wielkości bliżej wyników symulacji w porównaniu do wyników dla systemu $MAP/G/1/b$.

Wyniki analizy zostały porównane z wynikami dla systemu o niezależnych czasach usługi oraz z symulacjami bazującymi na śladach ruchu. Proponowany model $MAP/SM/1/B$ produkuje zawsze bardziej dokładne wyniki w porównaniu do poprzednich rozwiązań zakładających niezależne czasy obsługi. Jednakże stopień poprawy zmienia się znacząco w zależności od autokorelacji odstępów pomiędzy pakietami. W przypadku autokorelacji o dużym zasięgu, wpływ autokorelacji czasów obsługi na charakterystyki jest Podobna sytuację obserwuje się w wysoko obciążonych systemach. W przypadku umiarkowanego zasięgu autokorelacji odstępów pomiędzy pakietami poprawa dokładności po zastosowaniu modelu o skorelowanych czasach obsługi może sięgać kilku rzędów wielkości dla systemu o typowym rozmiarze bufora.

Przedstawione wyniki analityczne mogą pomóc w optymalizacji parametrów urządzeń dla różnych typów ruchu i dać wskazówki, jak poprawić ich konstrukcję, na przykład poprzez wprowadzenie mechanizmów kształtowania ruchu zmniejszającego negatywny wpływ autokorelacji długości pakietu na charakterystyki kolejek.

2 Budowa rutera

Interfejsy współczesnych routerów są skomplikowanymi konstrukcjami posiadającymi kilku buforów. Uwarunkowane jest to ograniczeniami zastosowanej pamięci lub chęcią uproszczenia systemu operacyjnego rutera. Struktura tych interfejsów jest bardzo kiepsko, jeżeli w ogóle udokumentowana. Jednocześnie znajomość wewnętrznej budowy urządzenia jest niezbędna w celu zbudowania poprawnego modelu matematycznego. Szczególnie istotna jest znajomość tzw. Strategii kolejkowania, czyli sposobu w jaki bufory przechowują pakiety.

Obecnie istnieją co najmniej dwie strategie przechowywania pakietów: strategia bajtowa i strategia pakietowa. Bufor o strategii bajtowej można przechować stałą ilość danych (w bajtach), tj. dużą liczbę krótkich pakietów lub małą liczbę długich pakietów. Z drugiej strony, w strategii pakietowej, bufor może pomieścić określoną liczbę pakietów niezależnie od ich wielkości. W pracach teoretycznych znacznie częściej spotyka się strategię bajtową np. [13, 4]. Niemniej jednak, modele pakietowe też są spotykane [10]. Przykładowo domyślna strategia w popularnym symulatorze ns-2 to strategia pakietowa [7].

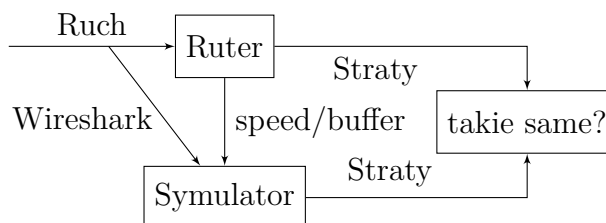
Podjęcie bajtowe jest poprawne dla niektórych typów routerów, gdzie pakiety są przechowywane w kawałkach pamięci [4], jednak nie jest odpowiednie dla wszystkich urządzeń. Jak zostanie pokazane zastosowanie poprawnej strategii ma ogromne znaczenie dla dokładności modelu interfejsu. W celu wyznaczenia strategii kolejkowania oraz innych parametrów, które nie zawsze są dobrze udokumentowane przeprowadzony został eksperyment na routerze CISCO klasy ISR. Wyniki eksperymentu pozwalają odpowiedzieć na następujące pytania:

1. Jak pakiety są kolejkowane ?
2. Jaki jest rozmiar kolejki ?
3. Jak dokładne jest przybliżenie interfejsu kolejką FIFO ?

Wyniki przedstawione w tej sekcji zostały opublikowane w [12].

2.1 Eksperyment

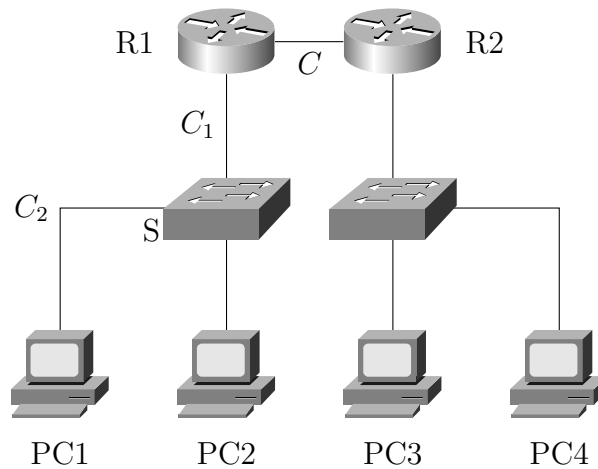
Eksperyment składał się z dwóch części. W pierwszym kroku Wygenerowany ruch był przesyłany przez stratny interfejs routera. Jednocześnie cały ruch był zapisywany na dysk. W drugim kroku przeprowadzone zostały symulacje z wykorzystaniem zebranych śladów ruchu w celu zweryfikowania, czy symulator i fizyczne urządzenie tracą te same pakiety. Koncepcja eksperymentu została schematycznie przedstawiona na rysunku. 1.



Rysunek 1: Concept of the experiment. We compared drops occurring in router and in trace-driven simulation

Każdy pakiet w swoim ładunku zawierał unikalne id, dlatego możliwa była identyfikacja każdej straty. Eksperyment był przeprowadzony w izolowanej sieci (rysunek 2), więc praktycznie cały ruch był generowany i zawierał znaczniki.

Do testów wykorzystano routery Cisco serii 2800 będące typowymi przedstawicielami rodziny ISR. Do wszystkich połączeń wykorzystano łącza ethernetowe 100 Mbit ($C_1 = C_2 = 100$) łącze do generowania strat miało szybkość $C = 10$ Mbit.



Rysunek 2: Siec do eksperymentu, R1 – testowany ruter, PC2 – generator ruchu, PC3 – komputer docelowy, PC1,PC4 – sniffery

2.2 Funkcja strat

W celu ilościowego opisu dokładności modelu wprowadzona została miara nazwana funkcją strat. Załóżmy, że M wejściowych pakietów N zostało utraconych zarówno w symulacji jak i na urządzeniu. Współczynnik strat będzie taki sam jednak wcale to nie znaczy, że symulator i urządzenie tracą pakiety w identyczny sposób. W celu dokładniejszego opisu procesu strat wprowadzona została funkcja strat $d(i), i \in \mathbb{N}$. Funkcja strat $d(i)$ przyjmuje dwie wartości:

$$d(i) = \begin{cases} 0 & \text{jeżeli } i\text{-ty pakiet nie został stracony} \\ 1 & \text{jeżeli } i\text{-ty pakiet został stracony} \end{cases} \quad (1)$$

I w pełni opisuje proces strat. Wykresy tej funkcji nie są przejrzyste, dlatego posługiwać się będziemy zakumulowaną funkcją strat zdefiniowaną następująco:

$$D(i) = \sum_{k=1}^i d(k). \quad (2)$$

Jest to liczba straconych pakietów od początku obserwacji do chwili przyjścia i -tego pakietu. Dla uproszczenia skumulowana funkcja strat będzie nazywana po prostu funkcją strat (FS).

Tak zdefiniowana FS jest wygodnym narzędziem służącym do porównywania strat w dwóch systemach. Jeżeli różnica FS dla tych systemów jest wszędzie równa zero dla każdego wejścia, to znaczy, że systemy te tracą pakiety w identyczny sposób.

W idealnym przypadku FS rutera (D_r) FS symulatora (D_s) powinny być takie same. W rzeczywistości zawsze pojawiają się błędy i niedokładności, jak również symulator może być zbyt uproszczony. Dlatego nie funkcje strat będą podobne, ale nie identyczne.

Aby zmierzyć dokładność symulatora(modelu) wykorzystana została różnica pomiędzy eksperymentalną i symulacyjną FS nazwana funkcją błędu

$$D_{\Delta}(i) = D_r(i) - D_s(i), \quad i \in \mathbb{N}. \quad (3)$$

2.3 Wyniki

Wyniki eksperymentów przeprowadzonych na fizycznych urządzeniach sugerowały, że kolejka na interfejsie wyjściowym jest znacznie większa niż ustalona w konfiguracji urządzenia. Dopiero uwzględnienie "ukrytych" buforów opisanych w [11] umożliwiło stworzenie poprawnego modelu urządzenia.

2.3.1 Budowa interfejsu rutera

Każdy interfejs rutera ma dwie kolejki: wejściowa i wyjściowa. Przycho-
dzący pakiet trafia najpierw do kolejki wejściowej, gdzie czeka na przetwo-
rzenie. W skrajnych przypadkach, kiedy ruter jest przeciążony obliczeniowo
(np z powodu dużej tablicy routingu) kolejka ta może się przepełnić i pakiet
jest tracony. Each interface in the router has an input and an output que-
ue. When a packet enters the router, it is queued in the input queue of the
incoming interface and waits to be processed.

Po przetworzeniu pakiet trafia do kolejki wyjściowej, gdzie czeka na wy-
słanie. Straty w kolejce wyjściowej są znacznie częstsze niż w wejściowej (ob-
serwowane są cały czas) i są powodowane przez przeciążenie rutera ruchem
(chwilowe wzrosty szybkości zgłoszeń).

Kolejka wyjściowa jest znacznie bardziej skomplikowana niż prosta ko-
lejką FIFO powszechnie używana jako jej model. Jest ona podzielona na
dwie kolejki: sprzętową(tx ring) i programową [11].

Pojemność kolejki programowej można zmieniać w Cisco IOS poleceniem
hold-queue podczas gdy kolejki sprzętowe są zarządzane przez system ope-
racyjny rutera. Dla sprzętowej kolejki można tylko sprawdzić jej rozmiar
poleceniem show controller. Wynik znajduje się w sekcji tx_limit. Pojem-
ność sprzętowej kolejki zależy od typu interfejsu. Ruter stara się ją zmini-
malizować ale i tak, jej pojemność może wynosić 128 pakietów dla interfejsu
FastEthernet.

Wszystkie zaawansowane mechanizmy kolejkowania są przeprowadzane
z wykorzystaniem kolejek programowych. Po skończeniu przetwarzania pa-

kietu, CPU umieszcza go w dzielonej sprzętowej kolejce wyjściowej docelowego interfejsu.

Jeżeli ta kolejka jest pełna to pakiety są przechowywane w kolejce programowej. Interfejs następnie wybiera pakiety z kolejki sprzętowej, kiedy jest gotowy do wysłania kolejnego pakietu. Jeżeli bufor sprzętowy jest pusty, interfejs przechodzi w tryb spoczynku.

W trybie spoczynku interfejs okresowo sprawdza czy w kolejce sprzętowej nie pojawił się pakiet. Jeżeli bufor zawiera nowy pakiet interfejs przechodzi do normalnego trybu i zaczyna jego przesyłanie.

W normalnym trybie interfejs zgłasza przerwanie do CPU, jeżeli potrzebuje więcej pakietów do wysłania. Powoduje to, że czas potrzebny na opuszczenie kolejki programowej jest nieokreślony i dlatego takiego systemu nie można modelować jako pary kolejek. *Można go jednak przybliżyć jedną kolejką, której pojemność równa jest sumie pojemności kolejki programowej i sprzętowej.* Takie przybliżenie okazuje się być bardzo dokładne.

2.3.2 Weryfikacja modelu

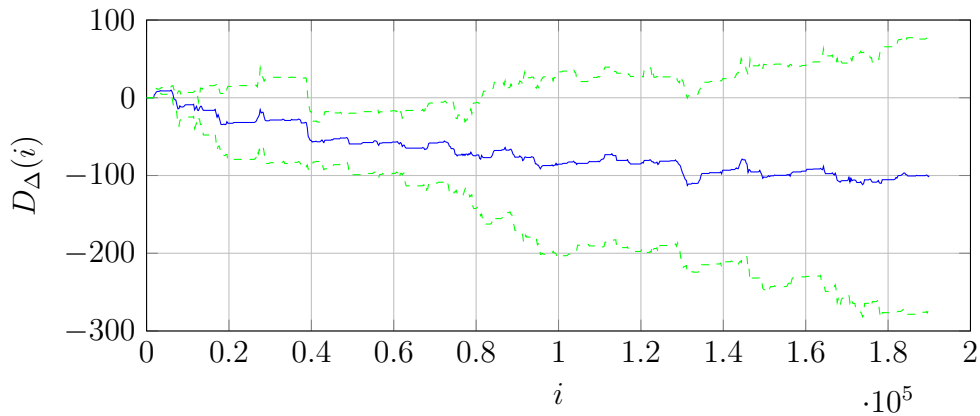
Wyniki przeprowadzonych eksperymentów pokazały, że połączenie kolejki programowej i sprzętowej w jedną jest dobrym przybliżeniem rzeczywistego urządzenia. Dodatkowo przeprowadzenie eksperymentów z różnymi rozmiarami pakietów pokazało, że wszystkie kolejki mają strategię pakietową.

Testy zostały przeprowadzone dla trzech różnych pojemności kolejki programowej na interfejsie Ethernet współczynnikiem obciążenia $\rho = 0.8$. Na rysunku 4 przedstawione zostały odpowiednie funkcje błędów oraz ich przedziały ufności. Każdy wykres przedstawia średnią funkcję błędów dla pięciu eksperymentów z identycznym ruchem wejściowym. Widać, że błąd jest bardzo losowy jednak wartość zerowa błędów prawie zawsze zawiera się w przedziale ufności.

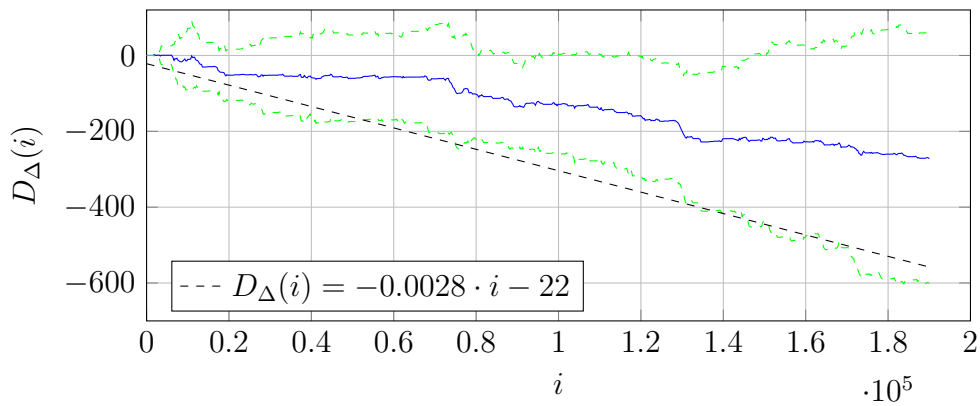
Pomimo wyraźnej losowości przedstawione funkcje błędów wykazują też pewne trendy. Taki trend może być spowodowany błędem w estymacji szybkości łącza lub pojemności kolejki. Zmieniając pojemność o ± 1 otrzymujemy znacznie gorszy model, dlatego można wnioskować, że trendy są spowodowane niedokładnym pomiarem szybkości interfejsu.

Zaobserwowana dokładność modelu nie zależy od pojemności kolejki programowej, jednak zależy od obciążenia. Rysunek 4 przedstawia wyniki otrzymane przy współczynnikiem obciążenia $\rho = 0.8$. Wyniki dla większych obciążeń (0.9 and 1.1) przedstawiono na rysunku 3.

Widać, że w przypadku większego obciążenia błędy są większe. Nasuwa się więc pytanie czy są to błędy akceptowalne. Aby na nie odpowiedzieć, rozważmy błąd pomiaru współczynnika strat w najgorszym przypadku z ry-



(a)



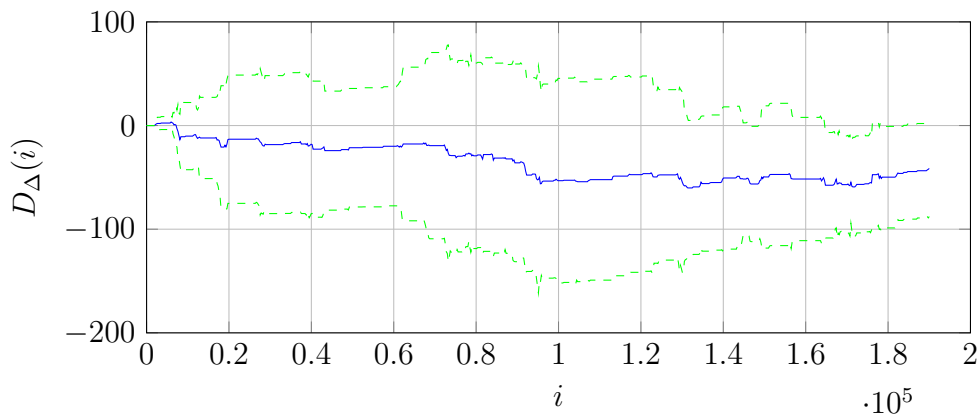
(b)

Rysunek 3: Funkcja błędu dla $\rho = 0.9$ (a), $\rho = 1.1$ (b), Kolejka programowa 20 pakietów. Linia ciągła – funkcja błędu, linia przerywana – przedział ufności 95%

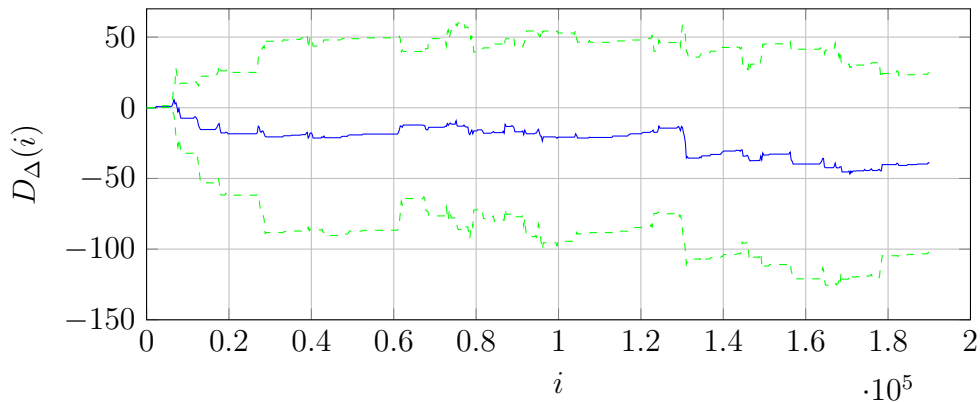
sunku 3. Funkcję błędu można przybliżyć linią prostą jak to pokazano na rysunku 3. Błąd współczynnika strat jest nachyleniem tej prostej i w naszym przypadku wynosi 0.0028 dla wartości współczynnika strat 0.189. Błąd ten nie jest bardzo duży stanowi jedynie 1.5 % wartości rzeczywistej dla silnie przeciążonego łącza. Dla typowych wartości obciążenia błąd jest znacznie mniejszy.

3 Model Matematyczny

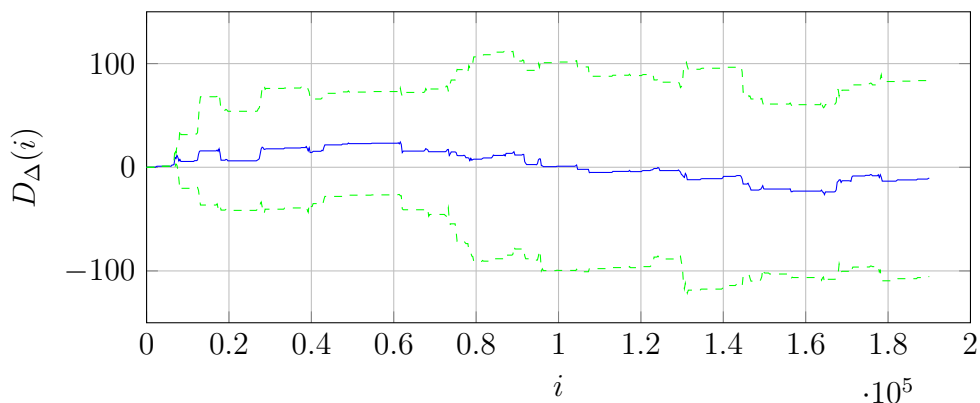
Interfejsy urządzeń sieciowych były modelowane na wiele sposobów. Większość modeli została opracowana z myślą o rządzeniach ATM i nie nadaje się



(a)



(b)



(c)

Rysunek 4: Funkcja błędu dla kolejki programowej: 20 (a), 40(b), 60(c) pakietów, $\rho = 0.8$. Linia ciągła – funkcja błędu, linia przerywana – przedział ufności 95%

do stosowania do stosowania w urządzeniach pracujących w sieci IP ponieważ pakiety IP mają różne długości.

W urządzeniach typu store and forward pakiet jest usuwany z kolejki jak tylko zostanie wysłany do przeznaczenia. Ponieważ czas potrzebny na przesłanie pakietu jest proporcjonalny do długości pakietu, proces obsługi w modelu analitycznym musi odzwierciedlać wszystkie własności statystyczne rozmiarów pakietów włączając ich autokorelację.

Ruch IP jest bardzo silnie i różnorodnie skorelowany. Odstępy czasu pomiędzy pakietami posiadają autokorelację, podobnie długości pakietów. Dodatkowo odstępy i długości mogą być wzajemnie skorelowane.

Pośród wielu podejść do modelowania ruchu sieciowego modele markowskie są szczególnie przydatne. Mimo iż nie są w stanie uwzględnić zjawisk takich samopodobieństwo to mogą je przybliżać na wielu skalach czasu co bardzo często jest wystarczające. Główna zaleta modeli markowskich jest to iż jedne z nielicznych pozwalają stworzyć modele systemów kolejkowych, które są rozwiązywalne analitycznie. Dlatego też ciągle są aktywnie rozwijane i badane [5, 3, 2, 1].

W związku z powyższym w proponowanym modelu interfejsu rutera markowski proces zgłoszeń (MAP) został wykorzystany jak model ruchu. Długości pakietów (a co za tym idzie proces obsługi) zostały modelowane jako Ukryte Modele Markowa (HMM) co zapewniło poprawne modelowanie autokorelacji długości pakietów. Ostatecznie zaproponowany model można zapisać jako MAP/SM/1/b korzystając z notacji Kendalla.

Dla takiego modelu zbadano proces strat. Wykorzystując metodę potencjału wyznaczone zostały charakterystyki takie jak: stacjonarny Współczynnik Strat czy chwilowa i stacjonarna szybkość strat. Zastosowana metoda potencjału sprawdziła się przy wyznaczaniu charakterystyk takich jak czas do przepełnienia bufora dla prostszych systemów. Dlatego rozszerzenie tych wyników dla systemu MAP/SM/1/b nie powinien stanowić problemu.

3.1 Notacja

Ponieważ proces wejściowy I czas obsługi mają charakter markowski Przed przedstawieniem głównych wyników wprowadźmy notację, która pozwoli uniknąć nieporozumień. Każdy parametr x procesu wejściowego (MAP) będzie oznaczany \check{x} podczas gdy odpowiadający mu parametr procesu obsługi będzie oznaczany \hat{x} . Symbol \oplus oznacza *bezpośrednią sumę macierzy* (nie mylić z sumą Kroneckera) – i jest uproszczeniem zapisem macierzy blokowo diagonalnej. Iloczyn Kroneckera będzie oznaczany \otimes .

Wektory kolumnowe jedynek I zer będą oznaczane $\mathbf{1}$ i $\mathbf{0}$. Tradycyjnie I oznacza macierz jednostkowa a I_x macierza jednostkowa o rozmiarze x .

3.2 Model ruchu

Model ruchu sieciowego powinien uwzględniać zarówno odstępny między pakietami jak długości pakietów. W podejściu wykorzystującym proces BMAP są one modelowane jako jeden proces. Jednak w przypadku kolejek o strategii pakietowej tracimy związek pomiędzy odstępami a długościami pakietów. Dlatego zostały one przedstawione jako niezależne procesy, co jest oczywiście pewnym przybliżeniem rzeczywistości. Jak się jednak okazuje nie jest to zbyt duże uproszczenie.

3.2.1 Proces zgłoszeń

Proces zgłoszeń pakietów to Markovian Arrival Process czyli MAP. Formalnie MAP jest definiowany jako dwuwymiarowy proces Markowa $(N(t), \check{J}(t))$ na przestrzenie stanów $\{(i, j) : i \geq 0, 1 \leq j \leq \check{m}\}$, z generatorem $(N(t), \check{J}(t))$ danym

$$Q = \begin{bmatrix} D_0 & D_1 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & D_0 & D_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & D_0 & D_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

gdzie D_k są macierzami $\check{m} \times \check{m}$. Elementy $D_k, k \geq 1$ są nieujemne, D_0 ma nieujemne element poza przekątną i ujemne na przekątnej. Dodatkowo $D = \sum_{k=0}^{\infty} D_k$ jest generatorem i $D \neq D_0$. $N(t)$ oznacza liczbę zgłoszeń w przedziale $(0, t]$ a $\check{J}(t)$ oznacza stan łańcucha modulującego o generatorze D .

Istnieje też tzw. Konstruktywna definicja MAP. Załóżmy, że modulujący łańcuch Markowa jest w stanie i . Po losowym czasie o rozkładzie wykładniczym z parametrem λ_i łańcuch zmienia stan. Z prawdopodobieństwem $p_i(0, j)$ następuje przejście do stanu j bez zgłoszenia. Z prawdopodobieństwem $p_i(1, j)$ następuje przejście do stanu j połączone z przyjściem zgłoszenia. Zakładamy, że :

$$p_i(0, i) = 0, \quad \sum_{k=0}^{\infty} \sum_{j=1}^{\check{m}} p_i(k, j) = 1 \quad (4)$$

i

$$\begin{aligned} \lambda_i &= -(D_0)_{ii}, & 1 \leq i \leq \check{m} \\ p_i(0, j) &= \frac{(D_0)_{ij}}{\lambda_i}, & 1 \leq i, j, i \neq j \leq \check{m} \\ p_i(k, j) &= \frac{(D_k)_{ij}}{\lambda_i} & 1 \leq i, j \leq \check{m}. \end{aligned}$$

Opisany proces potrafi bardzo dobrze naśladować zjawiska obserwowane w rzeczywistym ruchu sieciowym. Średnia intensywność zgłoszeń, która będzie potrzebna do wyznaczenia stacjonarnego współczynnika strat dana jest wzorem:

$$\lambda = \pi D \mathbf{1}, \quad (5)$$

gdzie π to rozkład stacjonarny dla D , czyli

$$\pi D = \mathbf{0}^T, \quad \pi \mathbf{1} = 1.$$

3.2.2 Długości pakietów

Jako model rozmiarów pakietów a co za tym idzie czasów obsługi przyjęty został Hidden Markov Model (HMM). Proces HMM może być też widziany jak process półmarkowski, czyli uogólnienie procesu Markowa w którym czas przebywania w stanach nie jest wykładniczy.

Ewolucją HMM rządzi łańcuch Markowa określony na przestrzeni stanów $S = \{1, \dots, \hat{m}\}$. Jednak stany tego łańcucha (stan w chwili t jest oznaczany przez \hat{j}_t) jest ukryty. W każdym stanie obserwowany jest tylko losowy wynik $l \in L = \{l_1, \dots, l_{\hat{d}}\}$.

Proces HMM jest charakteryzowany przez następujące parametry:

1. \hat{m} , liczbę stanów.
2. \hat{d} , liczbę różnych wyników w każdym stanie. Ten parametr określa liczbę różnych długości pakietów.
3. prawdopodobieństwa przejść pomiędzy stanami T :

$$T_{ik} = \mathbb{P}(j_{t+1} = k | j_t = i).$$

4. prawdopodobieństwa emisji E :

$$E_{ik} = \mathbb{P}(l_k \text{ at } t | j_t = i).$$

5. początkowy rozkład stanów π :

$$\pi_i = \mathbb{P}(j_1 = i).$$

Proces HMM umożliwia modelowanie autokorelacji długości pakietów. Jednocześnie dzięki swojej Markowskiej strukturze umożliwia tworzenie modeli systemów kolejkowych, z których można analitycznie wyznaczyć ich charakterystyki.

3.3 Interfejs jako system kolejkowy

W zaproponowanym modelu interfejs jest skończona kolejka FIFO z pojedynczym stanowiskiem obsługi. Pojemność systemu jest równa b i zawiera klienta aktualnie obsługiwanego.

Zakłada się, że serwer może być w jednym \hat{m} różnych stanów, które odpowiadają stanom HMM. W i -tym stanie czas obsługi ma rozkład $F_i(t)$, $i = 1, \dots, \hat{m}$. Po każdej obsłudze następuje przejście do nowego lub tego samego stanu z godnie z prawdopodobieństwami przejścia dla HMM.

Stan serwera w chwili t oznaczony jest $\hat{J}(t) \in S$. Zakłada się, że czas zaczyna się w chwili ukończenia obsługi.

3.4 Wyniki analityczne

Zacznijmy od długości kolejki w chwili t oznaczonej jako $X(t)$. Niech $L(t)$ oznacza liczbę strat zaobserwowanych w przedziale czasu $(0, t]$, a $\Delta_{n,i,j}(t)$ niech będzie jej średnią wartością warunkowaną początkową długością kolejki oraz stanami początkowymi MAP i SM:

$$\Delta_{n,i,j}(t) = \mathbb{E}(L(t) | X(0) = n, \check{J}(0) = i, \hat{J}(0) = j) \\ 0 \leq n \leq b, \quad 1 \leq i \leq \check{m} \quad 1 \leq j \leq \hat{m}. \quad (6)$$

Oznaczmy transformate Laplace $\Delta_{n,i,j}(t)$ przez $\delta_{n,i,j}(s)$, czyli:

$$\delta_{n,i,j}(s) = \int_0^\infty e^{-st} \Delta_{n,i,j}(t) dt. \quad (7)$$

W celu użycia notacji macierzowej $\delta_{n,i,j}(s)$ jest zamienione na wektor kolumnowy wykorzystując odwrotny porządek leksykograficzny :

$$\delta_n(s) = [\delta_{n,1,1}, \dots, \delta_{n,\check{m},1}, \dots, \delta_{n,1,\hat{m}}, \dots, \delta_{n,\check{m},\hat{m}}]^T, \quad (8)$$

i

$$\delta(s) = [\delta_0^T(s), \dots, \delta_b^T(s)]^T. \quad (9)$$

Drugą charakterystyką rozważaną w rozprawie jest czas do przepełnienia bufora oznaczony jako:

$$\tau_{n,i,j} = \inf\{t > 0 : X(t) = b | X(0) = n, \check{J}(0) = i, \hat{J}(0) = j\}. \quad (10)$$

The transform of the tail of $\tau_{n,i,j}$ is equal to:

$$h_{n,i,j}(s) = \int_0^\infty e^{-st} \mathbb{P}(\tau_{n,i,j} > t) dt. \quad (11)$$

Ponownie w celu użycia notacji macierzowej $h_{n,i,j}(s)$ jest zamienione na wektor kolumnowy wykorzystując odwrotny porządek leksykograficzny :

$$h_n(s) = [h_{n,1,1}(s), \dots, h_{n,\check{m},1}(s), \dots, h_{n,1,\hat{m}}(s), \dots, h_{n,\check{m},\hat{m}}(s)]^T. \quad (12)$$

W [1] korzystano z funkcji liczącej dla MAP zdefiniowanej następująco:

$$\check{P}_{i,j}(n, t) = \mathbb{P}(N(t) = n, \check{J}(t) = j | N(0) = 0, \check{J}(0) = i). \quad (13)$$

Ponieważ stan procesu obsługi tworzy dodatkowy stopień swobody dla warunków początkowych, nowa funkcja licząca została zdefiniowana następująco:

$$P_{ijkl}(n, t) = \mathbb{P}(N(t) = n, \check{J}(t) = k, \hat{J}(t) = l | N(0) = 0, \check{J}(0) = i, \hat{J}(0) = j). \quad (14)$$

Jest to tensor czwartego rzędu i w ogólnym przypadku jest trudny do wyznaczenia. Chyba, że początek czasu odpowiada chwili ukończenia obsługi i t jest czasem potrzebnym na obsługę jednego pakietu. W takim przypadku, zakładając niezależność procesów zgłoszeń i obsługi postać funkcji liczącej upraszcza się do:

$$P_{ijkl}(n, t) = T_{jl} \check{P}_{ik}(n, t). \quad (15)$$

Przed przedstawieniem głównych wyników wprowadźmy następującą notację:

$$A_{j,k}(s) = \int_0^\infty e^{-st} \check{P}(k, t) dF_j(t), \quad (16)$$

$$\bar{D}_{j,k}(s) = \int_0^\infty e^{-st} \check{P}(k, t) (1 - F_j(t)) dt, \quad (17)$$

$$\mathcal{A}_n = T \otimes I_{\check{m}} \cdot \bigoplus_{i=1}^{\hat{m}} A_{i,n}, \quad (18)$$

$$\mathcal{D}_k(s) = [\bar{D}_{1,k}^T, \dots, \bar{D}_{\hat{m},k}^T]^T \quad (19)$$

$$\bar{\mathcal{A}}_n = \sum_{k=n}^\infty \mathcal{A}_k \quad (20)$$

$$Y_k(s) = \left[\frac{\lambda_i p_i(k, j)}{s + \lambda_i} \right]_{ij} \quad (21)$$

$$\mathcal{Y}_k(s) = \bigoplus_{i=1}^{\hat{m}} Y_k(s). \quad (22)$$

Podstawowe wyniki teoretyczne pracy:

Twierdzenie 1. *W systemie MAP/SM/1/b, jeżeli wejściowy proces MAP is proces obsługi (SM) są niezależne i macierz T jest nieosobliwa to transformata Laplace czasu do przepelnienia bufopra ma postać:*

$$h_n(s) = \sum_{k=0}^{b-n} R_{b-k-n}(s) \mathcal{A}_k(s) h_{b-1}(s) - \sum_{k=1}^{b-n} R_{b-k-n}(s) C_k(s), \quad (23)$$

$$h_{b-1}(s) = W_b^{-1}(s) V_b(s), \quad (24)$$

gdzie

$$R_0(s) = \mathbf{0}, \quad R_1(s) = \mathcal{A}_0^{-1}(s), \quad (25)$$

$$R_{k+1}(s) = R_1(s) \left(R_k(s) - \sum_{i=0}^k \mathcal{A}_{i+1}(s) R_{k-i}(s) \right), \quad k \geq 1 \quad (26)$$

and

$$W_b(s) = (\mathcal{Y}_0(s) - I) \sum_{k=0}^b R_{b-k}(s) \mathcal{A}_k(s) + \mathcal{Y}_1(s) \sum_{k=0}^{b-1} R_{b-k-1}(s) \mathcal{A}_k(s), \quad (27)$$

$$V_b(s) = (\mathcal{Y}_0(s) - I) \sum_{k=1}^b R_{b-k}(s) C_k(s) + \mathcal{Y}_1(s) \sum_{k=1}^{b-1} R_{b-k-1}(s) C_k(s) - z(s), \quad (28)$$

$$C_n(s) = \sum_{k=0}^{n-1} \mathcal{D}_k(s) \mathbf{1}. \quad (29)$$

Twierdzenie 2. *Jeżeli wejściowy proces MAP is proces obsługi (SM) są niezależne i macierz T jest nieosobliwa to transformata Laplace średniej liczby strat w $(0, t]$ w systemie MAP/SM/1/b ma postać:*

$$\delta_n(s) = \left(R_{b-n+1}(s) \mathcal{A}_0(s) \sum_{k=0}^{b-n} R_{b-n-k}(s) B_k(s) \right) M_b^{-1}(s) l_b(s) + \sum_{k=0}^{b-n} R_{b-n-k}(s) g_k(b, s), \quad (30)$$

gdzie

$$B_n(s) = \mathcal{A}_{n+1}(s) - \bar{\mathcal{A}}_{n+1}(s) \bar{\mathcal{A}}_0^{-1}(s)$$

$$g_k(s) = \bar{\mathcal{A}}_{k+1}(s) \bar{\mathcal{A}}_0^{-1}(s) c_b(s) - c_{b-k}(s)$$

$$\begin{aligned}
l_b(s) &= \mathcal{Y}_1(s) \sum_{k=0}^{b-1} R_{b-k-1}(s)g_k(s) - (I - \mathcal{Y}_0(s)) \sum_{k=0}^b R_{b-k}(s)g_k(s) \\
M_b(s) &= (I - \mathcal{Y}_0(s)) \left(\sum_{k=0}^b R_{b-k}(s)B_k(s) + R_{b+1}(s)\mathcal{A}_0(s) \right) \\
&\quad - \mathcal{Y}_1(s) \left(\sum_{k=0}^{b-1} R_{b-k-1}(s)B_k(s) + R_b(s)\mathcal{A}_0(s) \right)
\end{aligned}$$

Główną zaletą metody potencjału jest mniejsza złożoność numeryczna, równa $\check{m}^3\hat{m}^3b^2$ podczas gdy złożoność dla metody bezpośredniej to $\check{m}^3\hat{m}^3b^3$.

Twierdzenia 1 i 2 tylko wartości $\delta_{n,i,j}(s)$. Aby wyznaczyć $\Delta_{n,i,j}(t)$ w zadanej chwili należy zastosować metody numerycznego odwracania transformaty Laplace'a np. [6]. Charakterystyki w stanie ustalonym można jednak wyznaczyć korzystając z własności granicznych transformaty Laplace

Przykładowo stacjonarny współczynnik strat można wyznaczyć następująco:

$$LR = \lim_{t \rightarrow \infty} \frac{1}{\lambda} \frac{d\Delta_{n,i,j}(t)}{dt} = \lim_{s \rightarrow 0^+} \frac{s^2 \delta_{n,i,j}(s)}{\lambda}. \quad (31)$$

3.5 Dyskretny czas obsługi

W praktycznych zastosowaniach twierdzeń 1 i 1 jest potrzeba wyznaczenia macierzy $A_k(s)$ and \bar{D} . Efektywna metoda ich wyznaczenia jest metoda uniformizacji [9]. W metodzie tej należy istotną rolę odgrywają następujące funkcjonały:

$$\gamma_j(s) = \int_0^\infty e^{-(s+\theta)t} \frac{(\theta t)^j}{j!} dF(t), \quad (32)$$

$$\beta_j(s) = \int_0^\infty e^{-(s+\theta)t} \frac{(\theta t)^j}{j!} (1 - F(t)) dt, \quad (33)$$

$$\theta = \max_i (-D_0)_{ii}.$$

Postać tych funkcjonałów jest szczególnie łatwo wyznaczyć dla dyskretnego rozkładu czasu obsługi jaki obserwuje w rzeczywistych systemach. Można pokazać, że postać ta jest następująca:

$$\gamma_j(s) = \sum_i p_i \frac{e^{-(\theta+s)d_i} (\theta d_i)^j}{j!}, \quad (34)$$

$$\beta_j(s) = \sum_i p_i (\theta^j \frac{\Gamma(j+1, 0) - \Gamma(j+1, d_i(s+\theta))}{j!(s+\theta)^{j+1}}), \quad (35)$$

Gdzie p_i to prawdopodobieństwo, że czas obsługi wyniesie d_i a $\Gamma(j, x)$ to niekompletna funkcja Gamma.

4 Testowanie modelu

Wszystkie wyniki numeryczne otrzymano dla rzeczywistego ruchu wygenerowanego przez akademik AGH w Krakowie. Przechwycone zostało 150000 pakietów IP na łączu wychodzącym z akademika. Do zebranych śladów dopasowano modele ruchu. Same ślady zaś zostały wykorzystane jako źródło ruchu w badaniach symulacyjnych, których wyniki zostały porównane z przewidywaniami modelu.

4.1 Estymacja parametrów

Parametry procesu MAP zostały wyznaczone za pomocą estymatora EM [8]. W modelu rzyjęto cztery stany modulujące ($\tilde{m} = 4$).

Parametry modelu HMM zostały wyznaczone za pomocą estymatora Bauma-Welcha zaimplementowanego w funkcji *hmmtrain* w programie MATLAB. Przed dopasowaniem modelu, rozmiary należy przetworzyć aby ograniczyć liczbę różnych długości pakietów, która dochodzi do setek.

W celu redukcji liczby długości pakietów zastosowany został algorytm k-średnich do zgrupowania długości w klastry. W praktyce pakiety są albo małe (≈ 50 B) albo duże (≈ 1500 B). Niewielki procent całości tworzy grupę średnią (≈ 600 B). Dlatego w trzy klastry wystarczają aby poprawnie modelować ruch.

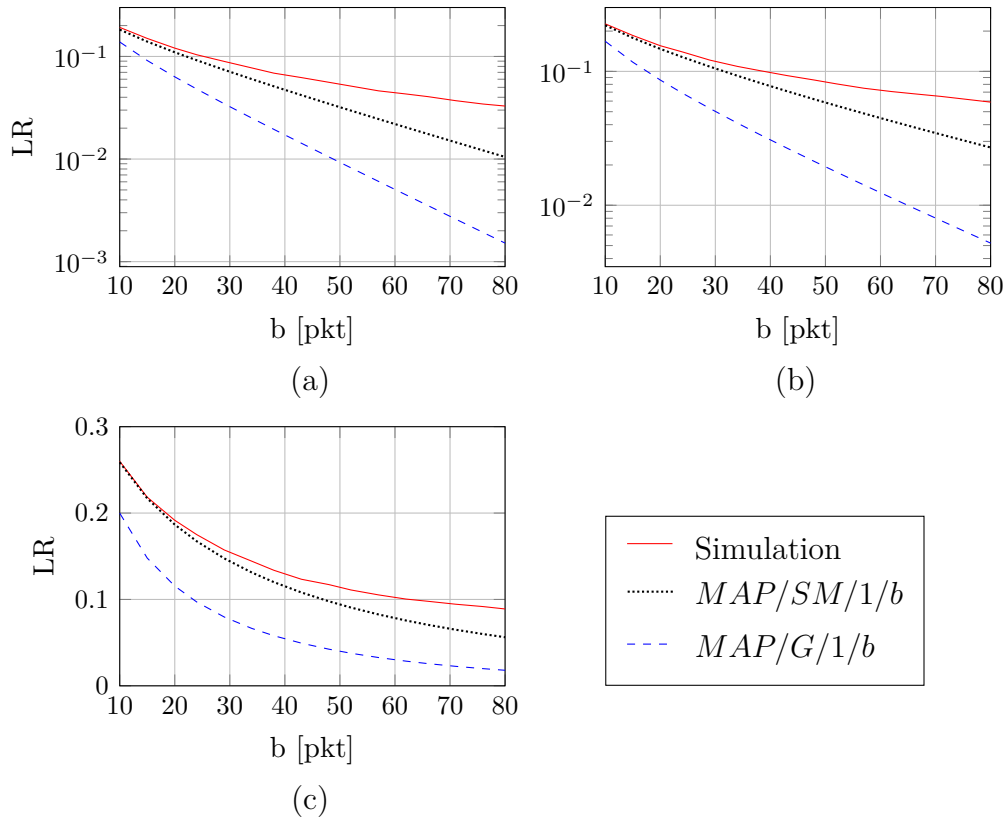
4.2 Wyniki numeryczne

Porównanie współczynnika strat dla wyników symulacyjnych i proponowanego modelu analitycznego przedstawiono na rysunku 5. Proces obsługi był modelowany jako czterostanowy HMM z dwoma rozmiarami pakietów ($\hat{d} = 2$).

Dla porównania do długości pakietów dopasowano też tradycyjny model niezależnych zmiennych losowych. Na rysunku 5 przedstawiono Współczynniki strat dla obydwu modeli.

Z porównania widać, że model ze skorelowanymi czasami obsługi jest znacznie bliższy rzeczywistości niż uproszczony model o niezależnych czasach. Obydwa modele tracą dokładność dla dużych rozmiarów bufora.

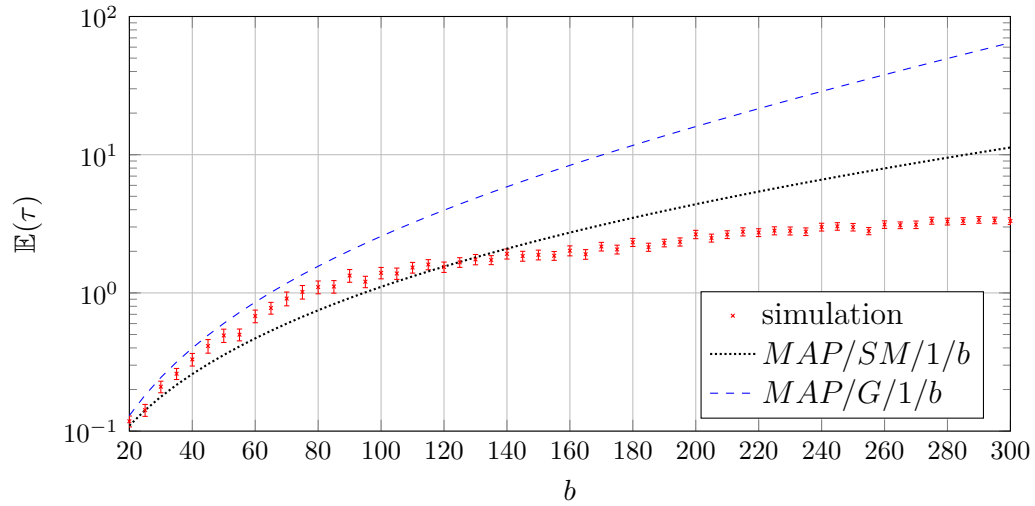
Pomimo tej niedogodności proponowany model jest znacznie bliższy rzeczywistości niż model z niezależnymi czasami obsługi. Jest to szczególnie widoczna dla dużych buforów gdzie różnica dochodzi do jednego rzędu wielkości. Jest to szczególnie istotne, jeżeli weźmiemy pod uwagę, dodatkowe bufory, które zwiększają całkowitą pojemność kolejki do ponad 150 pakietów.



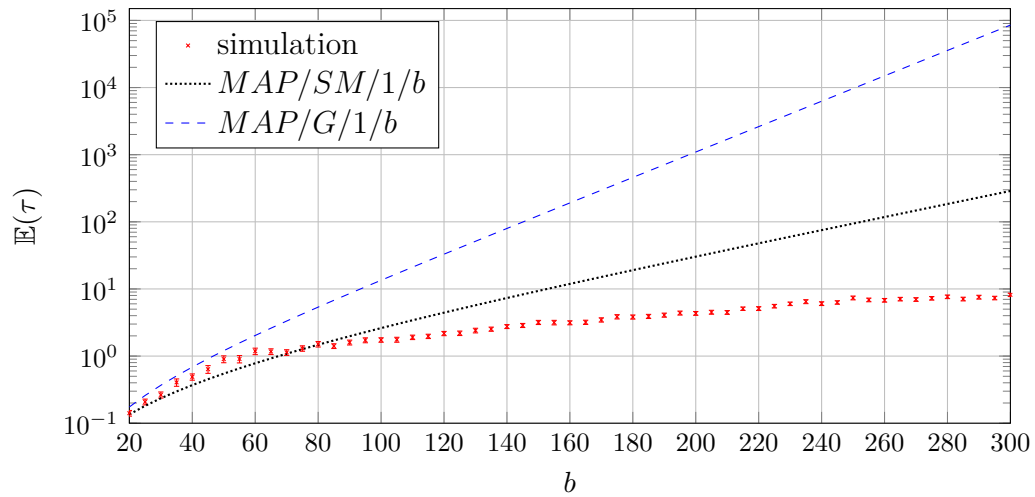
Rysunek 5: Stacjonarny współczynnik strat w funkcji pojemności bufora dla $\rho = 0.7$ (a), $\rho = 0.8$ (b) and $\rho = 0.9$ (c).

Wyniki dla stanu ustalonego wyznaczono korzystając własności granicznych transformaty Laplacea. Stosując algorytm numerycznego odwracania transformaty, można wyznaczyć charakterystyki przejściowe lub dokładną postać rozkładu czasu do przepełnienia bufora. Proponowany model jest jednym znanym modelem, który daje możliwość wyznaczenia charakterystyk nieustalonych dla skończonych kolejek o skorelowanych czasach obsługi.

Na rysunku 8 przedstawiono analizę jednej sekundy ruchu dla dwóch różnych konfiguracji początkowych. Natomiast rysunki 10 i 10 przedstawiają rozkład czasu do przepełnienia bufora. Widac, że dodanie autokorelacji czasu obsługi znacząco poprawia dokładność modelu. Analizując inne ślady ruchu i metody estymacji parametrów ruchu, stwierdzono, że dopiero dla ruchu o bardzo dużym zasięgu autokorelacji czasów pomiędzy zgłoszeniami, wpływ autokorelacji czasów obsługi jest pomniejszony.



Rysunek 6: Średni czas do przepełnienia bufora. Obciążenie $\rho = 0.95$, stany początkowe dane rozkładem stacjonarnym



Rysunek 7: Średni czas do przepełnienia bufora. Obciążenie $\rho = 0.80$, stany początkowe dane rozkładem stacjonarnym

5 Podsumowanie

Przeprowadzone badania przyniosły dwa istotne wyniki. Pierwszy to poprawny model interfejsu routera. Drugi to nowy wynik analityczny na postać procesu strat oraz rozkładu czasu do przepełnienia bufora w systemie $MAP/SM/1/b$.

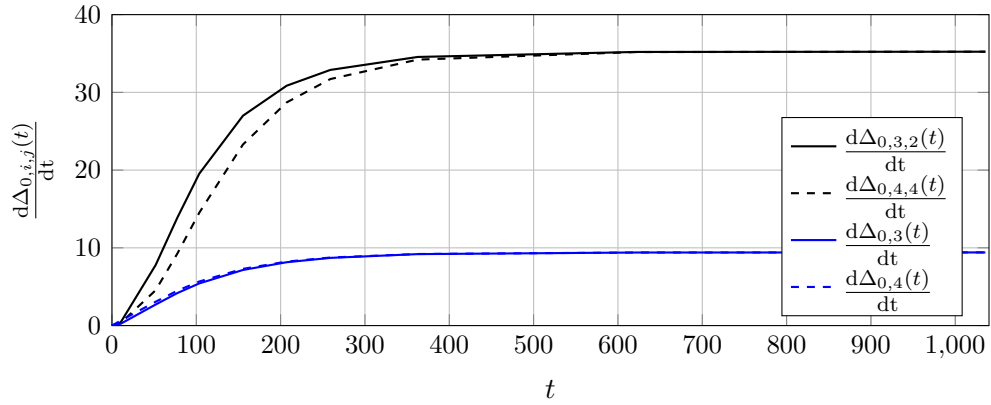
Najważniejsze osiągnięcia pracy można streścić w następujących punktach:

1. Wyczerpująca analiza stanu wiedzy na temat modelowania skończonych buforów.
2. Zaprojektowanie eksperymentu do zbadania mechanizmu kolejkowania w interfejsie routera.
3. Uogólnienie metody potencjału na systemy o skorelowanych czasach obsługi.
4. Procedury numeryczne do użycia mieszanek rozkładów w metodzie potencjału.
5. Wyprowadzenie zależności na proces strat w systemie $MAP/SM/1/B$.
6. Wyprowadzenie zależności na czas do przepełnienia bufora w systemie $MAP/SM/1/B$.
7. Implementacja procedur numerycznych do wyznaczania charakterystyk kolejek.
8. Porównanie dokładności $MAP/SM/1/B$ i $MAP/G/1/B$ jako modeli interfejsu routera.

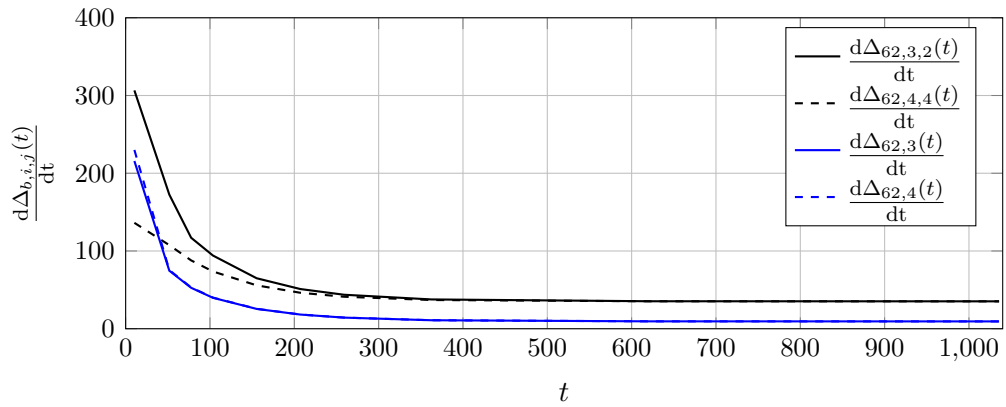
Literatura

- [1] A. Chydziański. Packet loss process in a queue with markovian arrivals. In *Networking, 2008. ICN 2008. Seventh International Conference on*, pages 524–529, april 2008.
- [2] Andrzej Chydziański. Queue size in a bmap queue with finite buffer. In Yevgeni Koucheryavy, Jarmo Harju, and Villy Iversen, editors, *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, volume 4003 of *Lecture Notes in Computer Science*, pages 200–210. Springer Berlin / Heidelberg, 2006.

- [3] Andrzej Chydziański. Time to reach buffer capacity in a bmap queue. *Stochastic Models*, 23:195–209, 2007.
- [4] Andrzej Chydziański and Ryszard Winiarczyk. Blocking probability in a bmap queue. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 547–553, Washington, DC, USA, 2006. IEEE Computer Society.
- [5] Alexander N. Dudin, Alexey A. Shaban, and Valentina I. Klimenok. Analysis of a queue in the bmap/g/1/n system. *International Journal of Simulation Systems, Science & Technology*, 6(1-2):13–22, 2005.
- [6] Abate Joseph, L. Choudhury Gagan, and Whitt Ward. An introduction to numerical transform inversion and its application to probability models. In W.K. Grassmann, editor, *Computational probability*, International series in operations research & management science, pages 257–323. Kluwer Academic, 2000.
- [7] K. Varadhan K. Fall. *The ns Manual*. The VINT Project, January 2009.
- [8] Alexander Klemm, Er Klemm, Christoph Lindemann, and Marco Lohmann. Modeling ip traffic using the batch markovian arrival process. *Performance Evaluation*, 54:149–173, 2003.
- [9] David M. Lucantoni. New results on the single server queue with a batch markovian arrival process. *Communications in Statistics - Stochastic Models*, 7(1):1–46, 1991.
- [10] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, and R. Lo Cigno. Markov models of internet traffic and a new hierarchical mmpp model. *Computer Communications*, 28(16):1835 – 1851, 2005.
- [11] Ivan Pepelnjak. Queuing principles in cisco ios. http://wiki.nil.com/Queuing_Principles_in_Cisco_IOS.
- [12] Krzysztof Rusek, Lucjan Janowski, and Zdzisław Papir. Correct router interface modeling. In *Proceedings of the second joint WOSP/SIPEW international conference on Performance engineering*, ICPE '11, pages 97–102, 2011.
- [13] Paulo Salvador, António Pacheco, and Rui Valadas. Modeling ip traffic: joint characterization of packet arrivals and packet sizes using bmaps. *Computer Networks*, 44(3):335 – 352, 2004.

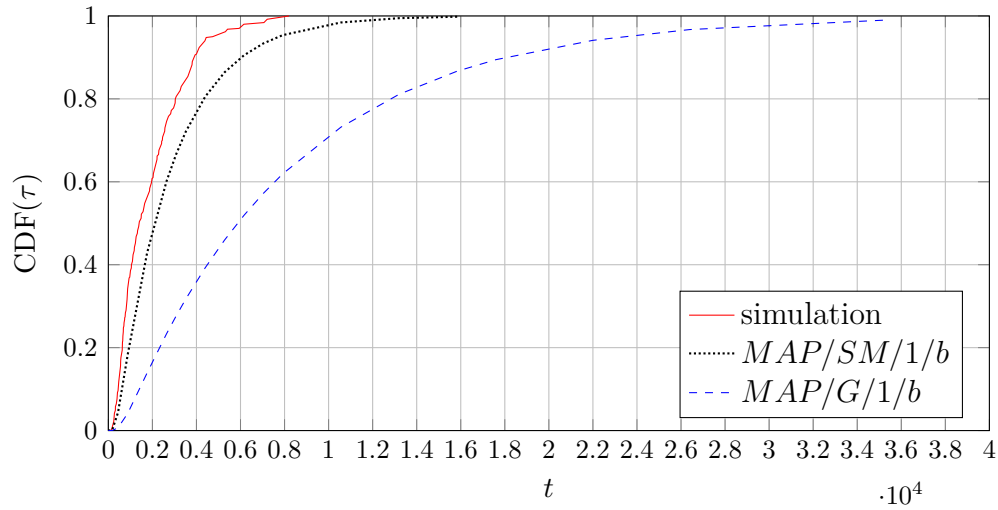


(a)

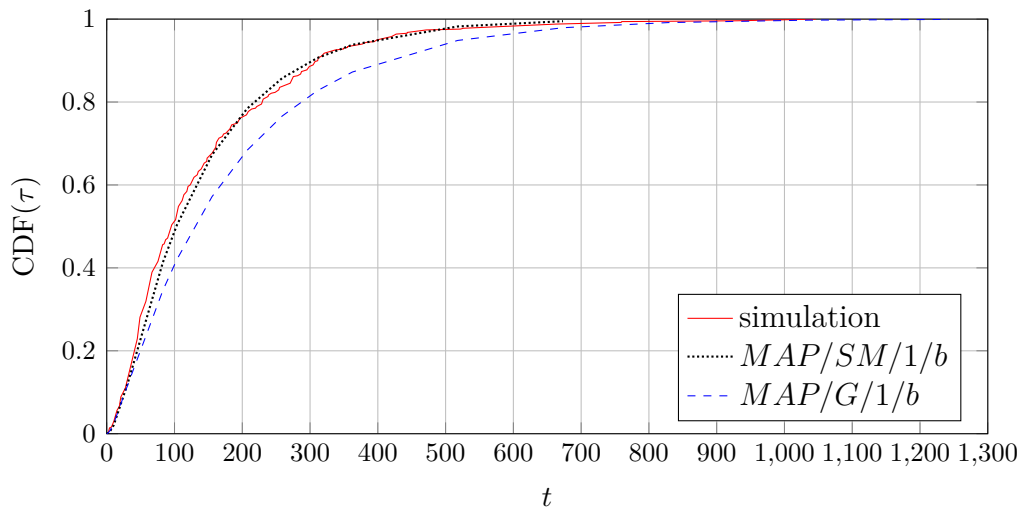


(b)

Rysunek 8: Chwilowa szybkość strat dla bufora początkowo pustego (a) i pełnego (b) ($b = 62$ pakiety). Jednostka czasu to średnia czas obsługi. Podwójnie indeksowana Δ to charakterystyka dla systemu $MAP/G/1/b$ (linia ciągła), Potrójnie indeksowana Δ to charakterystyka dla systemu $MAP/SM/1/b$ (linia przerywana).



Rysunek 9: Dystrybuanta czasu do przepełnienia bufora o rozmiarze 168 pakietów zaczynając od pustego systemu o obciążeniu $\rho = 0.95$.



Rysunek 10: Dystrybuanta czasu do przepełnienia bufora o rozmiarze 20 pakietów zaczynając od pustego systemu o obciążeniu $\rho = 0.8$