

Ryszard MACELUCH, Piotr KASPRZYK, Adam DOMAŃSKI

Politechnika Śląska, Instytut Informatyki

BADANIA MECHANIZMÓW QoS W SYSTEMIE LINUX

Streszczenie. Niniejszy artykuł opisuje badania mechanizmów zapewniania jakości usług dostępnych w systemie operacyjnym Linux. Przedstawione zostały badania routera programowego z zaimplementowanymi mechanizmami QoS. Pokazany został wpływ konfiguracji routera na kształtowanie ruchu w sieci.

Słowa kluczowe: sieci komputerowe, kształtowanie ruchu w sieci.

RESEARCH OF QoS MECHANISMS IN LINUX OPERATING SYSTEM

Summary. This article describe research of mechanisms of Quality of Service in operation system Linux. Research of software router with implemented QoS mechanism were described in paper. Influence of configuration of router on forming network traffic has been showed.

Keywords: computer network, shapping.

1. Wstęp

W niniejszym artykule zostaną opisane badania, których celem będzie ocena dostępnych aktualnie w systemie Linux mechanizmów QoS. Badania będą polegały na ocenie wpływu różnych czynników na zmienność opóźnień w przekazywaniu pakietów IP w routerze programowym (system operacyjny Linux). Badania przeprowadzone będą w trzech etapach. Pierwszy etap będzie dotyczył transmisji danych pomiędzy dwoma komputerami bez elementu pośredniczącego. Celem tych badań będzie sprawdzenie możliwości wysyłania strumienia danych TCP przez stację roboczą. Również możliwe jest pokazanie wpływu mechanizmów protokołów oraz systemu operacyjnego na zmienność opóźnień wysyłania pakietów w sieć. Drugi etap będzie dotyczył transmisji danych pomiędzy stacjami przy

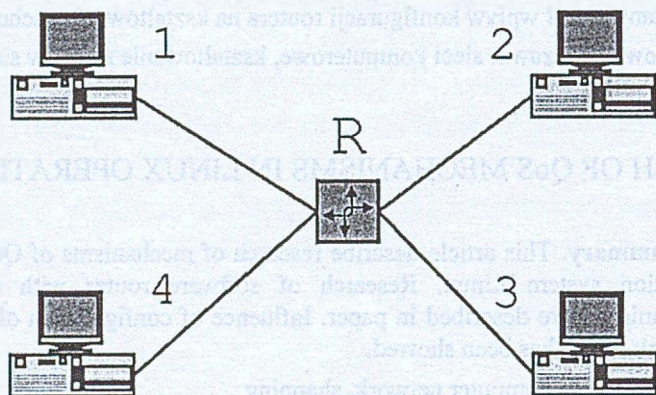
udziale elementu pośredniczącego (routera programowego bez włączonych mechanizmów QoS). Badania zostaną dokonane w warunkach zmiennego obciążenia routera. Etap trzeci będzie zrealizowany w identycznej konfiguracji pomiarowej jak etap drugi, przy włączonych mechanizmach QoS.

2. Metodyka badań

2.1. Sprzęt użyty w badaniach

Stanowisko badawcze (rysunek 1) składa się z następujących elementów:

- router programowy (komputer PC z systemem operacyjnym Linux),
- cztery stacje robocze (komputery PC z systemem operacyjnym Linux).



Rys. 1. Stanowisko badawcze

Fig. 1. Test configuration

Router programowy stanowi komputer klasy PC wyposażony w procesor Pentium 75 MHz, 48 MB pamięci operacyjnej oraz cztery karty sieciowe Ethernet 100 Mb/s. Komputer pracuje pod kontrolą systemu GNU/Linux Debian (jądro systemu w wersji 2.4.18). Połączenia ze stacjami roboczymi zrealizowano za pomocą skrętki kategorii 5. Stacje robocze to cztery komputery klasy PC Pentium III 733 MHz z pamięcią operacyjną 64 MB oraz pojedynczą kartą sieciową o przepustowości 100 Mb/s. Pracują one pod kontrolą systemu operacyjnego RedHat Linux w wersji 7.2 (jądro 2.4.18).

2.2. Opis programu testującego

W celu wykonania testów niezbędne jest zmierzenie opóźnienia występującego przy przekazywaniu pakietów IP. Należy również zmierzyć chwilową, średnią i graniczną wartości przepustowości. W systemie Linux można dokonać pomiarów w trybie użytkownika (ang. user mode) lub w trybie jądra (ang. kernel mode).

Tryb jądra to uprzywilejowany tryb dostępu do procesora. Proces działający w tym trybie może uczynić wszystko, co umożliwiała architektura sprzętowa, na której jest wykonywany. Nie jest ograniczany przez prawa dostępu do plików. Może odczytywać i zapisywać dane do przestrzeni adresowej innych procesów. Proces taki może być uruchomiony jedynie przez administratora danego komputera.

Trybem użytkownika określa się nieuprzywilejowany tryb dostępu do procesora. Proces taki podlega restrykcjom dotyczącym dostępu jedynie do własnej przestrzeni adresowej pamięci oraz jedynie do plików, z odpowiednimi prawami dostępu.

2.2.1. Pomiary w trybie użytkownika

Przykładowym rozwiązaniem działającym w trybie użytkownika jest wykorzystanie do pomiaru programu podsłuchującego ruch w sieci komputerowej (ang. sniffer). Należy uruchomić kilka instancji takiego programu (np. *tcpdump*), po jednej na każdy interfejs sieciowy. W związku z szeregowaniem procesów przez jądro, nie można zagwarantować w prosty sposób, że pomiary czasu dokonywane przez nasłuchujące programy będą dokładne. Wynika to z faktu, że jądro przydziela każdemu procesowi dostęp do procesora i po upływie pewnego czasu proces zostaje wywłączony z zasobu. Problem można rozwiązać, uruchamiając system Linux na maszynie wieloprocessorowej tak, aby każda instancja programu nasłuchującego pracowała na oddzielnym procesorze. W dostępnych warunkach laboratoryjnych realizacja takiego rozwiązania była niemożliwa. Dodatkowym mankamentem sposobu pomiaru czasu dokonanego w trybie użytkownika jest mniejsza dokładność niż w trybie jądra. Programy mogą odczytać czas systemowy z rozdzielczością 10^{-6} s, czyli znacznie mniej niż częstotliwość taktowania współczesnych procesorów. Ograniczenie to narzucone jest przez bibliotekę standardową języka C i związane z postulatem przenośności oprogramowania napisanego w języku C (na poziomie źródeł) na różnych platformach.

2.2.2. Pomiary w trybie jądra

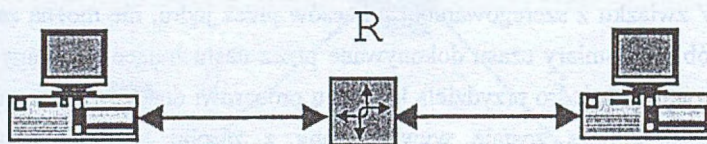
Wykonanie pomiarów w trybie jądra można przeprowadzić na dwa sposoby. Pierwszy z nich polega na modyfikacji źródeł, drugi wymaga napisania modułu jądra, które można załadować i wyładować dynamicznie w trakcie pracy systemu Linux. Po załadowaniu moduł traktowany jest jak integralna część jądra.

Mierząc czasy nadejścia pakietów z poziomu jądra można uzyskać pomiary z dokładnością do pojedynczego taktu zegara procesora. Np. dla routera z procesorem o częstotliwości 1 GHz osiągamy rozdzielczość czasu 10^{-9} s. Mankamenty omawianej metody związane są z nieprzenośnością programów oraz potencjalną niestabilnością systemu. Nieprzenośność spowodowana jest odczytem czasu z rejestru TSC, który występuje w procesorach Intel'a od modelu Pentium. Niestabilność z kolei wynika z braku kontroli systemu nad procesem działającym w trybie jądra. Zapis lub odczyt spod niewłaściwego adresu w pamięci może spowodować niestabilność całego systemu operacyjnego z jego zawieszeniem się włącznie. Utrudnia to testowanie modułu napisanego w trybie jądra w trakcie jego tworzenia. Po jego stworzeniu i przetestowaniu już nie odgrywa roli. W badaniach opisanych w tym artykule pomiary zostały dokonane w trybie jądra.

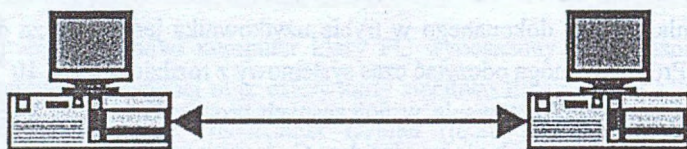
3. Opis wykonanych badań (bez mechanizmów QoS)

3.1. Pomiar prędkości przesyłu danych

Pomiarów dokonano w konfiguracjach przedstawionych na rysunkach 2 oraz 3. Czyli przy połączeniu stacji za pośrednictwem routera oraz przy połączeniu bezpośrednim.



Rys. 2. Połączenie przez router
Fig. 2. Connection through router



Rys. 3. Połączenie bezpośrednie
Fig. 3. Direct connection

Uzyskane prędkości transmisji przedstawia tabela 1. Wprowadzenie routera powoduje zmniejszenie chwilowych przeciążeń (poprzez buforowanie przychodzących pakietów) i tym samym „wygładzanie” ruchu. Widocznym efektem tego zjawiska jest zwiększenie prędkości transmisji w przypadku przesyłu danych poprzez router.

Tabela 1

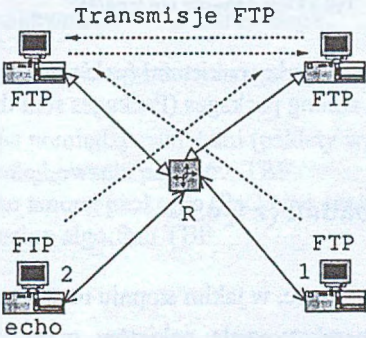
Prędkości transmisji uzyskane w czasie przesyłu danych

Rozmiar pliku [MB]	Połączenie bezpośrednie [B/s]	Połączenie przez router [B/s]
1	746956	742279
10	655519	754742
100	494265	633246

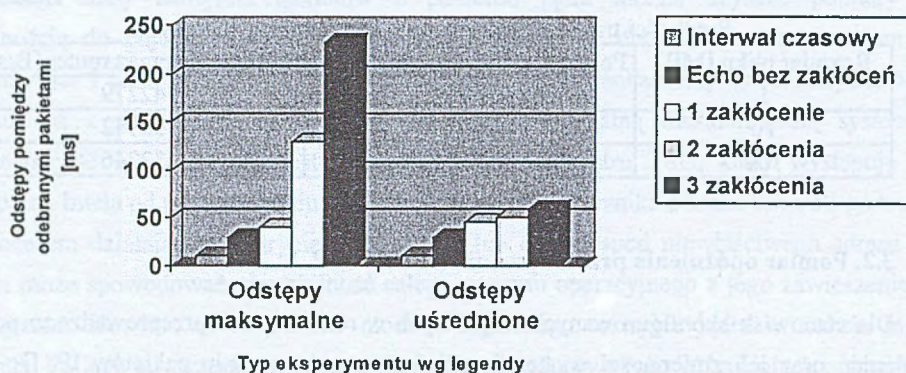
3.2. Pomiar opóźnienia przekazywania pakietów IP

Dla stanowisk skonfigurowanych i opisanych w rozdziale 3.1 przeprowadzono pomiary opóźnień oraz ich zmienności występujących przy przekazywaniu pakietów IP. Do badań wykorzystano program *echo* w wersji dla TCP udostępniany przez *inetd*. Działanie jego polega na powrotnym odsyłaniu pakietu docierającego do serwera. Pakiety wysyłane były seryjnie w równych, zadanych odstępach czasowych. Celem badań był pomiar odstępów pomiędzy odbieranymi pakietami TCP. Badania były wykonywane bez zakłóceń (w obecności transmisji *echa*) lub z zakłóceniami (przy realizacji przesyłu przez router dodatkowych transmisji wielomegabajtowych plików). Rysunek czwarty przedstawia schemat układu pomiarowego złożonego z routera oraz czterech stacji.

Rysunek 5 oraz 6 przedstawia wyniki pomiaru odstępów czasowych pomiędzy pakietami odebranymi. Na rysunku 5 zostało pokazane, jak zmieniały się odstępy pomiędzy przychodzącymi pakietami (pakiety były wysyłane co 10 ms) w przypadku zwiększania obciążenia routera. Rysunek 6 pokazuje analogiczną sytuację w przypadku gdy pakiety były wysyłane co 100 ms. Gdy obciążenie routera jest niewielkie lub nie istnieje, niezauważalny jest czas opóźnienia pakietu w routerze. Przy wzroście obciążenia routera rośnie czas obsługi pakietu w routerze.

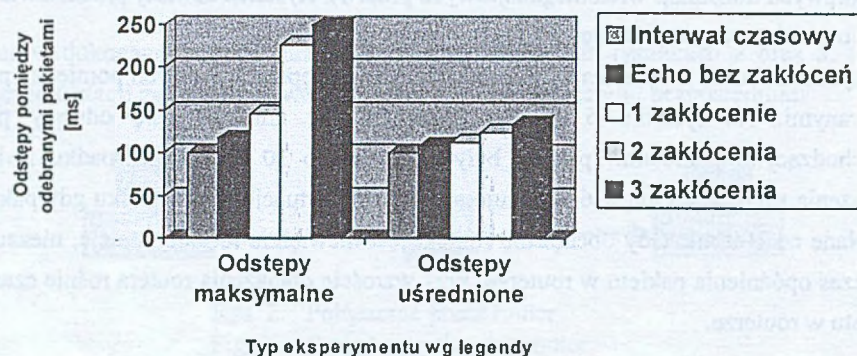


Rys. 4. Transmisja echo zakłócana przez cztery transmisje FTP
Fig. 4. Echo transmission disturbed by four FTP transmission



Rys. 5. Czas odstępu pomiędzy pakietami (pakiety wysyłane co 10 ms)

Fig. 5. Time of span among packages (Packages sent that 10 ms)



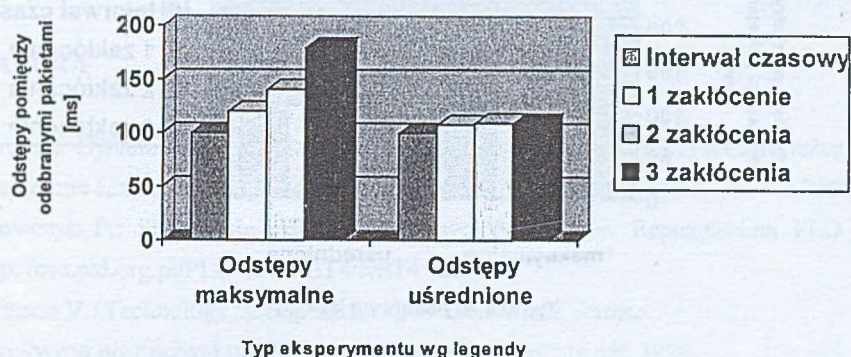
Rys. 6. Czas odstępu pomiędzy pakietami (pakiety wysyłane co 100 ms)

Fig. 6. Time of span among packages (Packages sent that 100 ms)

4. Opis wykonanych badań (z QoS)

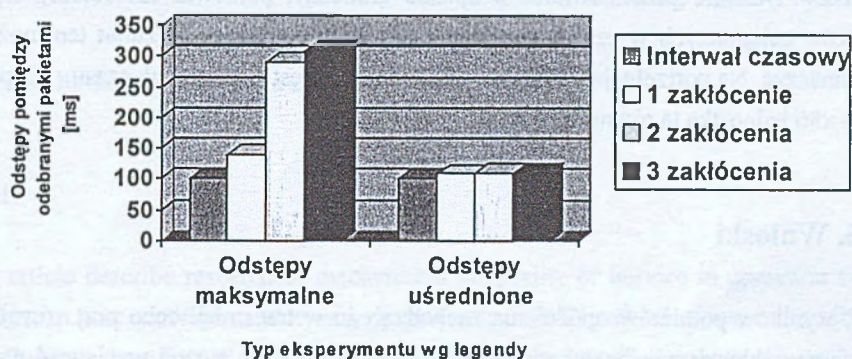
Badania miały na celu sprawdzenie, w jakim stopniu mechanizmy QoS mogą mieć wpływ na zmniejszenie opóźnienia przekazywania pakietów przez obciążony router. Punktem porównawczym była sytuacja pracy routera w przypadku braku obciążenia. W czasie badań wykorzystano algorytmy kolejowania pakietów PRIO, RED oraz TBF.

Uzyskane wyniki przedstawione zostały na rysunkach 7, 8 oraz 9.



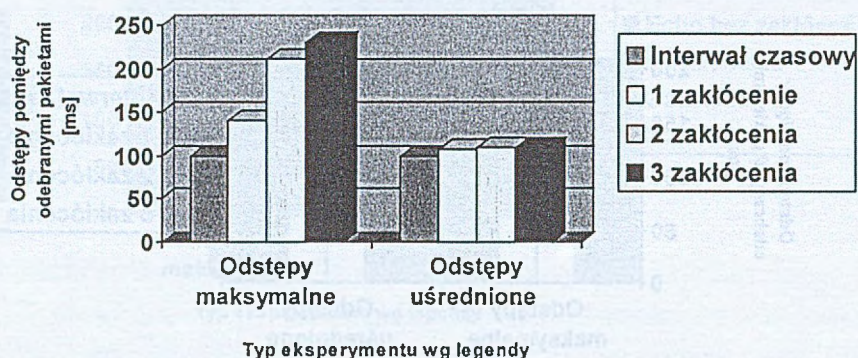
Rys. 7. Czas odstępu pomiędzy pakietami (pakiety wysyłane co 100 ms)
Algorytm kolejkowania pakietów PRIO

Fig. 7. Time of span among packages (Packages sent that 100 ms)
Packet queueing algorithm PRIO



Rys. 8. Czas odstępu pomiędzy pakietami (pakiety wysyłane co 100 ms)
Algorytm kolejkowania pakietów TBF

Fig. 8. Time of span among packages (Packages sent that 100 ms)
Packet queueing algorithm TBF



Rys. 9. Czas odstępu pomiędzy pakietami (pakiety wysyłane co 100 ms)
Algorytm kolejkowania pakietów RED

Fig. 9. Time of span among packages (Packages sent that 100 ms)
Packet queueing algorithm RED

W czasie prowadzenia badań wykonano również próbę wysyłania pakietów co 1 ms. Wyników tych nie przedstawiono w sposób graficzny, ponieważ nie różniły się one od wyników osiągniętych w czasie wysyłania pakietów co 10 ms. Rezultat ten można łatwo wytłumaczyć. Na potrzeby jądra Linuxa czas mierzony jest w kwantach czasu. Na platformie Intel'a x86 jednostka ta równa się 10 ms.

5. Wnioski

Z wyników pomiarów opóźnienia zachodzącego w transmisji echo przy równoczesnym transferze zakłócającym przechodzącym przez ten sam ruter można wyciągnąć następujące wnioski:

- Przy interwałach czasowych wysyłania pakietów mniejszych od 100 ms pojawiają się wartości mniejsze od zamierzonych. Wynika to ze sposobu w jaki ruter obsługuje pakiety TCP, tzn. jeśli pakiety nie zostają wysłane na czas z powodu przeciążenia, to później wysyłane są szybciej.
- „Wygładzanie ruchu” (uśrednianie opóźnień przekazywania pakietów) można zaobserwować głównie dla pakietów wysyłanych w większych odstępach czasowych. Wynika to z faktu, że jądro Linuxa na potrzeby własne odmierza czas jako wielokrotność jednostki równej 10 ms.

- Prosty algorytm kolejkowania PRIO dla naszego przypadku okazał się porównywalnie skuteczny jak algorytmy TBF oraz RED.

LITERATURA

1. Wilk K.: Dynamiczny przydział pasma użytkownika sieci z wykorzystaniem usługi QoS w systemie Linux. Praca dyplomowa magisterska, Politechnika Śląska, Gliwice 2002.
2. Krawczyk P.: Sterowanie przepływem danych w Linuxie. Repozytorium PLD CSV http://cvs.pld.org.pl/PLD-doc/NET4/NET4_tc.pdf.
3. Johnson V.: Technology Backgrounder-QoS-Glossary of Terms. http://www.ucer.nurcad.ufsc.br/docs/johnson_99technology.pdf, 1999.
4. Assured Forwarding PHB Group, RFC 2597, czerwiec 1999.
5. Shenker S., Patridge C., Guerin R.: Specification of Guaranteed Quality of Service. RFC 2212, wrzesień 1997.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 17 kwietnia 2003 r.

Abstract

This article describe research of mechanisms of Quality of Service in operation system Linux. Research of software router with implemented QoS mechanism were described in paper. Influence of configuration of router on forming network traffic has been showed.

Research have been carried in three periods: transmission beetwen workstation without router, transmission beetwen workstation through router, transmission beetwen workstation through router (with QoS). Figure 1, figure 2, figure 3 and figure 4 present test configurations. Time of span among packages was showed on figures 5, 6, 7, 8. Gotten results indicated that QoS mechanisms decrease packets latency and jitter.

Adresy

Ryszard MACELUCH: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, rmac@top.iinf.polsl.gliwice.pl.

Piotr KASPRZYK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, kwadrat@top.iinf.polsl.gliwice.pl.

Adam DOMAŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, adamd@top.iinf.polsl.gliwice.pl.