STUDIA INFORMATICA 2003
Volume 24 Number 2A (53)

Leszek BORZEMSKI, Krzysztof MUSIAŁ
Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów

ZARZĄDZANIE RUCHEM W ŚRODOWISKU SIECIOWYM LINUX

Streszczenie. W artykule została przedstawiona implementacja protokołów RSVP i MPLS na stosie TCP/IP w systemie Linux. Umieszczono krótką charakterystykę wybranych protokołów. Pokazano przykładową konfigurację w środowisku Linux. Wykonano eksperymenty pomiarowe pokazujące wykorzystanie tych protokołów do zarządzania ruchem w przypadku dostępu do serwera WWW.

Słowa kluczowe: zarządzanie ruchem, Linux, RSVP, MPLS.

TRAFFIC MANAGEMENT IN LINUX NETWORK ENVIRONMENT

Summary. This paper presents some practical aspects concerning implementation of traffic management functions based on Linux network environment. The main aim is to show how this functionality can improve access to WWW pages.

Keywords: traffic management, Linux, RSVP, MPLS.

1. Wprowadzenie

Rozwój systemu Linux na rynku sieci komputerowych trwa już od wielu lat. Początkowe implementacje systemów Linuxowych były niewielkimi wdrożeniami w małych sieciach, gdzie wykorzystywano te systemy głownie jako routery czy też serwery obsługujące pocztę lub serwisy WWW. Zainteresowanie tym systemem ciągle rośnie – głównie ze względu na jego cenę (jest to system rozprowadzany na zasadach licencji GNU) oraz dużą niezawodność. Ze względu na dużą popularność coraz większa liczba producentów oprogramowania aplikacyjnego decyduje się na implementacje swoich produktów na platformie Linuxa (jednym z pierwszych była firma Corel) – co zwiększa możliwości wykorzystania tego systemu jako m.in. profesjonalnej stacji graficznej.

Największą jednak popularność system ten do dzisiaj zawdzięcza swoim możliwościom pracy jako serwer (lub router) w sieciach komputerowych. Ma praktycznie wszystkie cechy pozwalające mu na implementacje większości rozwiązań i nowo powstających standardów. Znane są także implementacje tego systemu jako klastra serwerów (Mosix) do obsługi dużych ośrodków webowych z uwzględnieniem współpracy z bazami danych. System serwera WWW Apache na platformie Linuxa jest obecnie najpopularniejszym oprogramowaniem tego typu na świecie. Linux zawiera od dawna funkcje umożliwiające zarządzanie kontami użytkowników z ograniczeniami wykorzystania przestrzeni dyskowej (Quota), które w systemach konkurencyjnych (takich jak np. Microsoft Windows) pojawiły się dopiero w ostatnich latach. System ten obsługuje też najnowsze standardy sieci komputerowych, zarówno w środowisku sieci LAN, jak też MAN i WAN. Oferuje wszystkie znane zabezpieczenia współczesnych sieci komputerowych, jakimi są m.in. ściany ogniowe (firewall), pozwala na efektywne kształtowanie i zarządzanie ruchem – a implementacje iproute2 czy iptables są uważane za jedne z efektywniejszych w obecnych systemach komputerowych. Linux [1] jest również wyposażony w funkcje zarządzania sieciami komputerowymi w oparciu o protokoły RSVP (Resource ReSerVation Protocol) [4, 5, 6] i MPLS (MultiProtocol Label Switching) [2, 7]. Celem niniejszej pracy jest przedstawienie wybranych praktycznych aspektów implementacji zarządzania ruchem z wykorzystaniem protokołów RSVP i MPLS w sieci systemów Linux oraz eksperymentalne pokazanie wpływu konfiguracji połączeń sieciowych na jakość współpracy z serwisem WWW.

2. Zarządzanie ruchem w sieciach komputerowych

Dynamiczny rozwój sieci Internet, jaki zapoczątkowała usługa informacyjna WWW, która stała się w niedługim czasie podstawową usługą aplikacyjną dostarczaną w sieci, oraz jej rozwój i zmiany potrzeb użytkowników wymusza na producentach systemów sieci komputerowych wprowadzanie nowych rozwiązań sieciowych. Dotyczy to również potrzeb jakościowych usługi WWW. W ramach usługi WWW przetwarzane są i przesyłane różne rodzaje danych (obiekty). Pozyskiwanie zasobów odbywa się aktualnie bez rozróżnienia potrzeb transportowych poszczególnych rodzajów obiektów. Wszystkie zasoby są udostępniane na takich samych warunkach. Oznacza to, że nie ma możliwości określenia pożądanej zindywidualizowanej jakości obsługi transmisji. Parametry takiego „równouprawnienia" zostały określone w czasie podstawowych założeń przy projektowaniu sieci Internet. W sieci takiej dostęp do pasma danych miał być sprawiedliwy dla wszystkich użytkowników sieci (aplikacji sieciowych).

Rozwój nowych usług i zmiana wymagań użytkowników sieci, wzrost ilości przesyłanych danych oraz ich znaczenia (systemy bankowe itp.) wymaga zapewnienia przesłania danych z określonymi parametrami. W rozwoju takich usług ma pomóc rozwój sieci Internet drugiej generacji, nazywanej również „Internetem szerokopasmowym". W założeniach projektowych określono między innymi wsparcie dla takich usług, jak videokonferencje, VoIP, systemy B2B, B2C, szerokiego spektrum usług obliczeniowych, strumieniowych etc.

W chwili obecnej podstawowym problemem transmisji w sieci Internet jest zapewnienie odpowiednich parametrów jakości przekazu informacji na całej ścieżce transmisyjnej klient – serwer, czyli tzw. transmisji *QoS* (*Quality of Service*) *end-to-end*. Opracowano trzy koncepcje zróżnicowania obsługi pakietów IP z gwarancjami jej jakości, a mianowicie usługi zintegrowane (*IntServ*) - zdefiniowane w ścisłym powiązaniu z protokołem RSVP służącym do rezerwacji zasobów w sieciach IP; usługi zróżnicowane (*DiffServ*) - w których klasyfikacja pakietów odbywa się w węźle wprowadzającym dane do sieci oraz usługę wieloprotokołowej komutacji etykietowania MPLS, która integruje cechy protokołu IP i techniki ATM, wprowadzając m.in. możliwość doboru trasy przesyłania poszczególnych pakietów wg kryteriów istotnych dla zapewnienia jakości usługi QoS.

2.1. Protokoły zarządzania ruchem

IntServ

Architektura usług *Integrated Services* jest oparta na rezerwacji zasobów w sieci. Aby zapewnić odpowiednie parametry przekazu danych poprzez sieć, aplikacja musi zarezerwować pasmo przed rozpoczęciem transmisji. Wymaga to przeprowadzenia kilku działań. Po pierwsze, należy określić parametry ruchu oraz zapotrzebowanie na pasmo. Następnie protokół routingu sieci określa ścieżkę odpowiadającą żądanym parametrom. W kolejnym kroku następuje zarezerwowanie pasma w punktach na trasie przekazu. Po zarezerwowaniu pasma może nastąpić transmisja danych. Znane są dwie koncepcje realizacji proponowanego rozwiązania: *guaranteed service* i *controlled load service*. Pierwsza metoda wpiera aplikacje czułe na opóźnienia, druga zaś równoważy zapotrzebowanie na pasmo i opóźnienie występujące w sieci.

Model *IntServ* był pierwszą próbą zapewnienia parametrów jakości transmisji *QoS* w sieci Internet. Rozwój tego modelu sieci nie był jednak zbyt dynamiczny. Wynikało to z kilku powodów: przeznaczony był dla aplikacji o długim czasie działania i wrażliwych na opóźnienia pakietów w sieci. Szybko rozwijająca się wówczas usługa WWW opierała się raczej na modelu krótkich transakcji, w związku z tym czas rezerwacji pasma był relatywnie zbyt długi, aby implementacja usługi *RSVP* mogła przynieść znaczące korzyści. Kolejnym problemem jest mała skalowalność takiego rozwiązania. Usługi *IntServ* wykorzystywane

mogą być w sieciach o niedużym zasięgu np. w sieciach korporacyjnych dla zapewnienia parametrów videokonferencji , VoIP.

DiffServ

Podstawową różnicą usług zróżnicowanych *Differentiated Services* w stosunku do rezerwacji pasma w usługach zintegrowanych jest wykorzystanie kombinacji: zarządzania krawędziowego, zabezpieczenia i zarządzania priorytetami, aby uzyskać zróżnicowanie usług. W architekturze *DiffServ* ruch użytkownika jest „podzielony" na klasy. Dla każdej klasy ilość przekazywanych danych do sieci jest ściśle określona i nadzorowana na „brzegu sieci". Krawędź sieci jest zdolna do mapowania pakietów zgodnie z odpowiednią dla nich klasą ruchu. Klasyfikacja pakietów przeważnie ustalana jest na podstawie kontraktu SLA – *Service Level Agrement*, pomiędzy użytkownikiem i providerem sieci.

Typ klasy kodowany jest w nagłówku pakietu, podczas „wprowadzania" pakietu do sieci. Na podstawie klasy pakietu wewnętrzne urządzenia sieci rozróżniają sposób traktowania takiego pakietu. Usługi *DiffServ* nie wymagają rezerwacji pasma, a zasoby wymagane dla danej klasy są określone parametrami kontraktu SLA. Nie mamy tutaj też do czynienia z przekazem pojedynczych pakietów, a raczej z agregacją wszystkich pakietów danej klasy do pojedynczego strumienia. Usługi te eliminują problemy skalowalności występujące w przypadku *IntServ*. Funkcje obsługi pakietów wewnątrz sieci są proste, a jedynie decyzje podczas wprowadzania pakietów do sieci mogą być bardziej złożone.

MPLS

Technika używana w protokole *MPLS* jest znana jako *label switching*, gdzie krótka etykieta jest kodowana w nagłówku pakietu i używana do przekazywania pakietu. Podczas przetwarzania oznakowanego pakietu LSR – *Label Switch Router* używa etykiety do określenia kolejnego przeskoku w sieci. Używając etykietowania pakietów, określamy ścieżkę nazywaną *Label Switched Patch*. MPLS integruje cechy protokołów *IntServ* i *DiffServ* zapewniając rezerwację pasma na ścieżce LSP oraz oznaczania pakietów, w których może nastąpić zmiana priorytetu.

3. Eksperyment

Celem eksperymentu było pokazanie możliwości wykorzystania protokołów zarządzania ruchem uwzględniających dzisiejsze zapotrzebowanie użytkowników sieci.

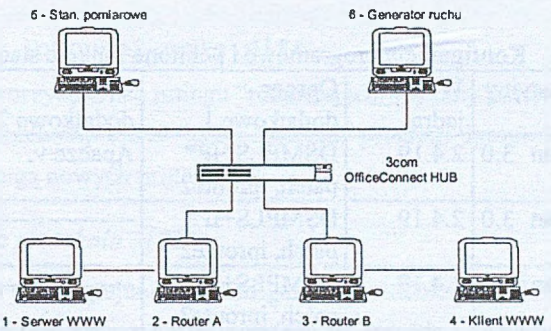
Przygotowanie i wykonanie eksperymentu podzielone zostało na kilka etapów:

- przygotowanie stanowiska badawczego,
- omówienie implementacji protokołów i konfiguracji w systemie Linux,

- wykonanie badań,
- prezentacja i omówienie wyników.

3.1. Przygotowanie stanowiska badawczego

Stanowisko badawcze zostało przygotowane z wykorzystaniem sześciu stanowisk komputerowych oraz huba jako punktu centralnego sieci. Tabela 1 zawiera opis parametrów technicznych stanowisk.



Rys. 1. Schemat stanowiska badawczego
Fig. 1. The testbed configuration

Tabela 1

Parametry techniczne stanowisk

Nr stanowiska	Procesor	Pamięć	Dysk twardy	Karta sieciowa
1-5	Celeron 300	128MB	20GB	Rtl 8139 10/100
6	Celeron IV 1,7GHz	256MB	20GB	Rtl 8139 10/100
HUB	3Com OfficeConnect 10Base HUB			

Każde ze stanowisk spełnia w naszym badaniu określoną funkcję. Stanowisko nr 1 jest to serwer WWW, na którym umieszczona została specjalnie przygotowana strona WWW, zawierająca kilka obiektów, służąca jako wzorzec pomiarowy. Stanowiska nr 2 i 3 są to routery pośredniczące w przekazywaniu informacji – zostały one zastosowane w celu zobrazowania rzeczywistej sieci komputerowej, w której następuje przekazywanie pakietów pomiędzy poszczególnymi „hopami". Stanowisko nr 4 to klient WWW, czyli przeglądarka, której zadaniem jest wysłanie żądania pobrania strony i obiektów na niej umieszczonych, ich odebranie i zobrazowanie na ekranie monitora. Stanowisko nr 5 – to stanowisko pomiarowe, na którym został zainstalowany system operacyjny Windows 98 oraz pakiet „Domino NAS" firmy Acterna Inc. Jest to analizator protokołów, który służył nam do analizy i oceny wydajności pomiarów. Funkcją stanowiska nr 6, na którym został zainstalowany generator ruchu (z pakietu „Domino NAS)", była symulacja obecności innych użytkowników w sieci i

„zapychanie" przez nich dostępnego pasma. Jak widać, stanowisko to zostało wyposażone w najmocniejszy procesor. Wynika to z ograniczeń programowych generatorów ruchu, których wydajność (szybkość generowania pakietów) ściśle zależy od mocy obliczeniowej procesora. W naszym przypadku (kanał 10 Mb/s) dopiero procesor o powyższych parametrach był w stanie generować pakiety z szybkością mogącą zapełnić całą dostępną przepustowość łącza. Jako punkt centralny zastosowano ośmioportowy koncentrator firmy 3Com z serii OfficeConnect o przepustowości 10 Mb/s. Zastosowanie koncentratora 10, a nie 100 Mb/s wynikało z ograniczeń wydajności programowego generatora ruchu.

Tabela 2

Konfiguracje programowe i pełnione funkcje stacji

Nr stacji	System operacyjny	Wersja jądra	Oprogr. dodatkowe 1	Oprogr. dodatkowe 2	Pełniona funkcja
1	Linux Debian 3.0 Woody	2.4.19	DSMPLS+IP* patch, iproute2	Apacze v.	Serwer WWW
2	Linux Debian 3.0 Woody	2.4.19	DSMPLS+IP* patch, iproute2	----------	Router
3	Linux Debian 3.0 Woody	2.4.19	DSMPLS+IP* patch, iproute2	----------	Router
4	Linux Debian 3.0 Woody	2.4.19	DSMPLS+IP* patch, iproute2	XWidnows(KDE) + Mozilla	Klient WWW
5	Windows 98 SE	4.10.2222	Domino NAS**	----------	Analizator protokołów
6	Windows XP Prof.	5.1.2600	Domino NAS**	----------	Generator ruchu
*	Pim Van Heuven z INTEC Broadband Communication Networks				
**	Analizator protokołów firmy Acterna, Inc.				

3.2. Implementacja i konfiguracja protokołów MPLS i RSVP w systemie Linux

W punkcie tym przytoczymy podstawowe informacje dotyczące implementacji i uruchomienia protokołów wymaganych do przeprowadzenia eksperymentu.

W eksperymencie wykorzystano stacje komputerowe z zainstalowanym systemem Linux, dystrybucja Debian 3.0 Woody. Ze względu na wymagania dodatkowych modułów wykorzystaliśmy stabilną wersję jądra nr 2.4.19 [8].

W tym miejscu należy zauważyć, że aplikacje wymagające obsługi sieci wysyłają odpowiednie żądanie do jądra systemu, którego zadaniem jest przygotowanie i wysłanie pakietów zgodnie z zakładanym standardem i typem sieci, w której dany system pracuje. W podstawowej konfiguracji jądra system Linux obsługuje protokoły TCP/IP, nie posiada natomiast implementacji protokołów RSVP oraz MPLS. W pierwszym kroku musimy zatem do stosu protokołów systemu dodać wymagane dla naszego eksperymentu poprawki.

3.2.1. Konfiguracja jądra systemu

Do tego celu wykorzystano patch DSMPLS+IP do jądra 2.4.19, autorstwa Pim Van Heuvena z instytutu INTEC Broadband Communication Networks na Uniwersytecie GENT [1, 10]. Umieszczamy odpowiedni plik w katalogu zawierającym źródła jądra Linux (standardowo /usr/src/linux) i wykonujemy polecenia aktualizacji kodu źródłowego jądra

```
patch -p0 < DSMPLS+IP.patch                                    (1)
```

W katalogu ze źródłami programu iproute2 (/usr/src/iproute2) dokonujemy aktualizacji poleceniem

```
patch -p1 < iproute2-mpls.diff.                                (2)
```

Dzięki temu możemy tworzyć tablice rutingu "rozumiejące" etykiety protokołu MPLS.

Następnie należy przekompilować jądro systemu, program iproute2 [9] i uruchomić ponownie system z obsługą nowych protokołów.

3.2.2. Konfiguracja protokołu MPLS

W następnym kroku przystępujemy do konfiguracji protokołu MPLS. W celu dokładnego zobrazowania przykładu przedstawimy prostą konfigurację systemu na końcowych stanowiskach pomiarowych nr 1 i 4.

Konfiguracja na stanowisku nr 1

```
mplsadm -A -O 0                                                (3)
mplsadm -O 0x02 -o push:gen:17:set:eth0:ipv4:10.0.200.2
ip route add 10.0.200.2/32 via 10.0.100.1 lsp 0x02
mplsadm -A -I gen:16:0
mplsmadm -L eth0:0
```

Konfiguracja na stanowisku nr 4

```
mplsadm -A -O 0                                                (4)
mplsadm -O 0x02 -o push:gen:16:set:eth0:ipv4:10.0.100.1
ip route add 10.0.100.1/32 via 10.0.200.2 lsp 0x02
mplsadm -A -I gen:17:0
mplsmadm -L eth0:0
```

Po wykonaniu powyższych poleceń cały ruch generowany do określonych miejsc w sieci zostaje umieszczony w pakiecie MPLS z odpowiednio przyporządkowaną etykietą. Poprawność konfiguracji możemy sprawdzić przeglądając dane systemowe w katalogu /proc/net, - są to odpowiednio pliki mpls_in, mpls_out, mpls_labelspace, w których przechowywane są informacje o wysłanych i odebranych pakietach i rezerwacji etykiety dla określonego połączenia. Oczywiście, generowany ruch można przejrzeć również za pomocą analizatora protokołów, w którym widać „opakowany" pakiet IP w nagłówek MPLS.

3.2.3. Konfiguracja protokołu RSVP

Do poprawnej realizacji protokołu RSVP musimy skonfigurować interfejsy sieciowe do obsługi rezerwacji zasobów uruchamiając skrypt

```
is_config etykieta_interfejsu,                    (5)
```

a następnie użyć dostępnego narzędzia, jakim jest demon RSVP. Uruchamiamy go poleceniem (na wszystkich stanowiskach)

```
rsvpd -D                                          (6)
```

Po uruchomieniu demona przystępujemy do rezerwacji zasobów dla określonych połączeń (na konsoli demona RSVP):

Stacja nr 1 (wejściowa -ingress)

```
dest lsp tcp 10.0.200.2/12                        (7)
sender 10.0.100.1/12
```

Stacja nr 4 (wyjściowa – egress)

```
dest lsp tcp 10.0.200.2/12                        (8)
reserve 10.0.200.2 ff 10.0.100.1/12
```

Następnie wykonujemy polecenie na stacji nr 1 (z konsoli systemowej)

```
tunel -L -c                                       (9)
```

i otrzymujemy w odpowiedzi informację o rezerwacji kanału

```
LSPID  Destination (type label/exp/iface)  viface  Packets  Bytes   (10)
12     10.0.200.2  (gen 21650/ 0/ eth0)    T21650  0        0
```

Ostatnim krokiem jest zamapowanie typu pakietów, które mają być przekazywane zarezerwowanym kanałem:

```
tunnel -m -p tcp -d 10.0.200.2/32 -l 12           (11)
```

Po wykonaniu powyższych poleceń wszystkie pakiety przekazywane pomiędzy stacjami o adresach IP 10.0.100.1 i 10.0.200.2 protokołem TCP zostaną przesłane zarezerwowanym kanałem RSVP. Należy zauważyć, że w tym przypadku nie ma znaczenia, która stacja jest stacją wejściową (*ingress router*), a która wyjściową (*egress router*), ponieważ i tak wszystkie pakiety określonego typu pomiędzy nimi zostaną przekazane zarezerwowanym kanałem. Konfiguracja ingress/egress router określa tylko, która ze stacji zażądała zestawienia kanału.

Należy również zwrócić uwagę na fakt, że implementacja protokołu MPLS nie jest wymagana poprzez żaden ze standardów, a wynika z konstrukcji implementacji powyższego rozwiązania, które można definiować zarówno dla usług zintegrowanych „Integrated Services", do której należy protokół RSVP, jak i dla usług zróżnicowanych „Differentiated Services", które wymagają usługi MPLS do agregacji ruchu. Z punktu widzenia wydajności

takiego systemu jest to o tyle niedobre, że następuje powiększenie pakietu o nagłówek MPLS, co zwiększa rozmiar danych nadmiarowych podczas przesyłania takich pakietów poprzez sieć.

3.3. Badania i omówienie wyników

Pokazanie wpływu protokołu RSVP/MPLS na czasy pobrania strony WWW wymagało realizacji pomiarów w trzech konfiguracjach:

- pobranie strony przy nieobciążonej sieci (konfiguracja TCP 0%),
- pobranie strony przy obciążonym paśmie (konfiguracja TCP 97%),
- pobranie strony w kanale MPLS/RSVP przy obciążonym paśmie konfiguracja MPLS/RSVP).

Serwer i klient WWW zostały uruchomiono odpowiednio na stanowiskach nr 1 i 4. Po każdym badaniu opróżniony był cache przeglądarki, aby zawsze następowało pobranie obiektów z serwera, a nie z lokalnego dysku klienta. Podczas ostatniego badania uruchomiony został generator pakietów zapełniający pasmo w ok. 97%. Przygotowana strona składała się ze szkieletu i czterech obiektów: obiekt 1 (szkielet strony) o rozmiarze 749 bajtów, obiekt 2 (658531 B), obiekt 3 (664206 B), obiekt (643537 B) i obiekt 5 (587859 B).

Na rysunkach 2, 3 i 4 przedstawiono czasy pobrania strony i obiektów. Tabela 3 przedstawia wyniki pomiarów średnich czasów pobrania stron w badanych konfiguracjach.

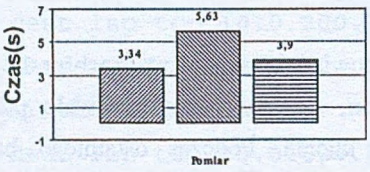
Tabela 3

Czasy dostępu do strony WWW i jej obiektów

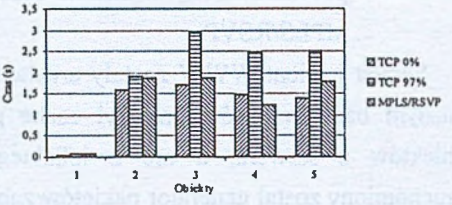
Obiekt	Rozmiar	TCP 0%			TCP 97%			MPLS/RSVP		
		Czas [s]			Czas [s]			Czas [s]		
		rozp.	zak.	całk.	rozp.	zak.	całk.	rozp.	zak.	całk.
Szkielet	749	0.00	0.04	0.04	0.00	0.06	0.06	0.00	0.06	0.06
Obiekt 2	658531	0.14	1.73	1.59	0.95	2.87	1.92	0.15	2.01	1.86
Obiekt 3	664206	0.14	1.85	1.71	0.10	3.04	2.94	0.14	2.01	1.87
Obiekt 4	643537	1.76	3.25	1.49	3.01	5.46	2.45	2.05	3.30	1.25
Obiekt 5	857859	1.93	3.34	1.41	3.14	5.63	2.49	2.12	3.90	1.78
Całość	2554882	0.00	3.34	**3.34**	0.00	5.63	**5.63**	0.00	3.90	**3.90**

Jak można zauważyć z wykresu na rys. 2, czas pobrania całej strony był najkrótszy w warunkach kanału, który nie był dodatkowo dociążany. Pobranie strony w warunkach dużego obciążenia pasma było wyraźnie wolniejsze, natomiast wykorzystanie przy transmisji protokołu rezerwacji pasma znacznie skróciło czas dostępu. Możemy również zauważyć, że czas pobrania strony w przypadku wolnego kanału i wykorzystania rezerwacji zasobów jest

niewielki. Można w tym wypadku wysnuć dwa przypuszczenia. Po pierwsze, zarezerwowany kanał na poziomie 7,5 Mb/s był w zupełności wystarczający do przesłania informacji, a po drugie – niewielki wzrost mógł być spowodowany wzrostem ilości nadmiarowej (nagłówków pakietów MPLS i RSVP) przekazywanej informacji. Na rys. 3 pokazano czasy pobrania w różnych warunkach transmisyjnych dla poszczególnych obiektów znajdujących się na stronie. W tym przypadku widać, że nasze oczekiwanie potwierdziło się – najkrótszy czas pobrania obiektu jest w pustym kanale – chociaż w przypadku pobrania obiektu nr 2 różnice były zdecydowanie mniejsze niż procentowa różnica pobrania całej strony, a w przypadku obiektu nr 4 możemy zauważyć, że najkrótszy był czas pobrania obiektu przy zarezerwowanym kanale.

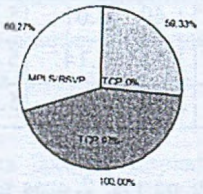


Rys. 2. Porównanie czasu pobrania strony WWW w różnych konfiguracjach
Fig. 2. Web page transfer time for different tests



Rys. 3. Porównanie czasu pobrania obiektów w różnych konfiguracjach
Fig. 3. Object transfer time for different tests

Na podstawie powyższych wyników możemy określić procentowy zysk czasu potrzebnego na pozyskanie wymaganych przez użytkownika sieci obiektów w przypadku znacznego obciążenia kanału transmisyjnego przy wykorzystaniu protokołów MPLS/ RSVP. Jako punkt odniesienia przyjęliśmy pomiar w warunkach maksymalnego obciążenia kanału, jako że jest to najczęściej spotykany przypadek w rzeczywistych sieciach komputerowych. Uzyskany wzrost wydajności w przypadku rezerwacji kanału jest na poziomie ok. 30%, co wydaje się wynikiem zadowalającym.



Rys. 4. Procentowy zysk wydajności przy wykorzystaniu protokołów MPLS/RSVP
Fig. 4. Performance increasing when using MPLS/RSVP (in percentages)

4. Podsumowanie

Rozwój nowych usług w sieciach komputerowych wymagających ściśle określonych zasobów i parametry sieci Internet skłaniają do opracowywania metod i protokołów pozwalających na lepsze wykorzystanie kanału transmisyjnego i umożliwienie zapewnienia parametrów jakości transmisji w danym kanale. W pracy pokazano praktyczną implementację takiego rozwiązania w szybko zdobywającym popularność i rynek systemie operacyjnym, jakim jest Linux. Wykorzystanie protokołu MPLS oraz RSVP umożliwiło zarezerwowanie pasma w mocno obciążonym kanale i uzyskanie przepływu danych na „dobrych" warunkach. Zastosowanie takich rozwiązań jest bardzo szerokie, poczynając od sieci lokalnych, w których nawet szerokopasmowy GigaEthernet nie ma możliwości rezerwacji zasobów, a kończąc na globalnej sieci Internet, w której dominującym protokołem jest TCP/IP, którego usługi są wykorzystywane do przenoszenia informacji o rezerwacji kanału. Dzięki takim rozwiązaniom klienci usługodawców internetowych mogą uzyskać zapewnienie o jakości połączenia w szeroko wykorzystywanych systemach transakcyjnych, co gwarantuje ciągłość sesji i tym samym podnosi bezpieczeństwo pracy takiego użytkownika, czy też w coraz popularniejszych transmisjach video, takich jak telekonferencje, VoIP itp. Dzięki wykorzystaniu do tego celu darmowego systemu, jakim jest Linux, można niewielkimi nakładami finansowymi uzyskać bardzo dobre efekty wydajności i wykorzystania kanałów transmisyjnych w dzisiejszych, mocno obciążonych, sieciach komputerowych.

LITERATURA

1. Van Heuven P., Van der Berghe S., Coppens J., Deemester P.: RSVP-TE demon for DiffServ over MPLS under Linux, INTEC – University of Gent, Linux Congress 2002.
2. Stankiewicz R., Jajszczyk A.: Wieloprotokołowa komutacja etykietowa (MPLS) i jej rola w zapewnieniu jakości usług w sieciach IP, Katedra Telekomunikacji AGH w Krakowie, Przegląd Telekomunikacyjny 2002, nr 4.
3. Stankiewicz R., Jajszczyk A.: Sposoby zapewnienia gwarantowanej jakości usług w sieciach IP, Katedra Telekomunikacji AGH w Krakowie, Przegląd Telekomunikacyjny 2002, nr 2.
4. RFC 2205 – Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.
5. RFC 2210 - The Use of RSVP with IETF Integrated Services.
6. RFC 3209 - RSVP-TE: Extensions to RSVP for LSP Tunnels.
7. RFC 3270 - Multi-Protocol Label Switching (MPLS) Support of Differentiated Services.

8. Linux Kernel 2.4.19 – http://ww.kernel.org.
9. Iproute2 v. 2.2.4 – ftp://ftp.inr.ac.ru/ip-routing/.
10. DSMPLS+IP-patch – http://dsmpls.atlantis.rug.ac.be/.
11. MPLS-Linux-1.172 – http://sourceforge.net/projects/mpls-linux/.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 7 kwietnia 2003 r.

Abstract

The purpose of this work is to show how MPLS and RSVP protocols are implemented in Linux operating system as well as to show how they can improve transfer of web pages. The short description of appropriate network technologies for providing quality of service is given. In the second part, the configuration procedure for activating both protocols in Linux environment is presented. Next, the testbed is presented and the results of the measurements are discussed.

Adresy

Leszek BORZEMSKI: Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, leszek@ists.pwr.wroc.pl.
Krzysztof MUSIAŁ: Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, kmusial@ists.pwr.wroc.pl.