STUDIA INFORMATICA 2002
Volume 23 Number 2A (48)

Krzysztof GROCHLA, Przemysław GŁOMB
Instytut Informatyki Teoretycznej i Stosowanej PAN

KSZTAŁTOWANIE RUCHU W RUTERACH PRACUJĄCYCH W SYSTEMIE LINUX

Streszczenie. Artykuł ma na celu zaprezentowanie możliwości kształtowania ruchu udostępniane przez system Linux umożliwiające zachowanie zasad jakości usług QoS w sieciach wykorzystujących rutery pracujące w tym systemie. Zaprezentowano najczęściej stosowane metody kolejkowania pakietów oraz możliwości ich wykorzystania do filtrowania ruchu. Omówiono także zbudowaną w IITiS PAN testową sieć komputerową oraz przeprowadzone w niej testy wybranych kilku metod.

TRAFFIC SHAPING IN ROUTERS WORKING ON LINUX

Summary. The article presents possibilities of traffic shaping build in Linux. Described functions allows to maintain the Quality of Service (QoS) in networks based on Linux routers. The second chapters enumerates a few most widely used queuing disciplines and packet filtering methods. The last part contains the description of the test network installation build in IITiS PAN and some experiment made there.

1. Wprowadzenie

Obecnie transmisje w sieciach komputerowych mają bardzo zróżnicowany charakter. Zwiększenie dostępności sieci Internet pozwala na uruchamianie coraz bardziej zaawansowanych usług, o bardziej zróżnicowanych wymaganiach co do parametrów transmisji. Dlatego konieczne staje się gwarantowanie użytkownikowi określonego poziomu jakości usług (Quality of Service). Osiąga się to poprzez kształtowanie ruchu (ang. *traffic shaping*) rozumiane tutaj jako opóźnianie transmisji bądź odrzucanie wybranych pakietów mające na celu spełnienie zadanych parametrów transmisji [1]. Jako klasę ruchu rozumiemy

transmisję danych tego samego typu (np. HTTP, FTP). W opracowaniu też często używano zamiennie sformułowań: metoda kolejkowania oraz algorytm kolejkowania, oznaczających sposób umieszczania pakietów oczekujących na transmisję w buforze.

Sieć Internet wykorzystuje do transmisji danych protokół IP, używany także w sieciach lokalnych. Na przestrzeni ostatnich kilku lat prowadzone są intensywne prace nad wprowadzeniem mechanizmów gwarantowania QoS do tego protokołu.

Ze względu na koszty często jako ruter w niewielkich sieciach wykorzystuje się nie dedykowane urządzenie, lecz odpowiednio oprogramowany komputer PC. Jako system operacyjny takich urządzeń szczególnie dużą popularność zdobył system Linux, ponieważ jest bezpłatny oraz stosunkowo łatwy w konfiguracji. Dzięki nowym możliwościom kolejkowania pakietów zaimplementowanym w jądrze systemu w wersji 2.2 lub nowszych istnieje możliwość zagwarantowania uzytkownikom określonej jakości usług transmisji danych. Poprzez odpowiednie skonfigurowanie opisanych niżej mechanizmów można np. zagwarantować określone pasmo dla wybranego komputera czy określonej usługi.

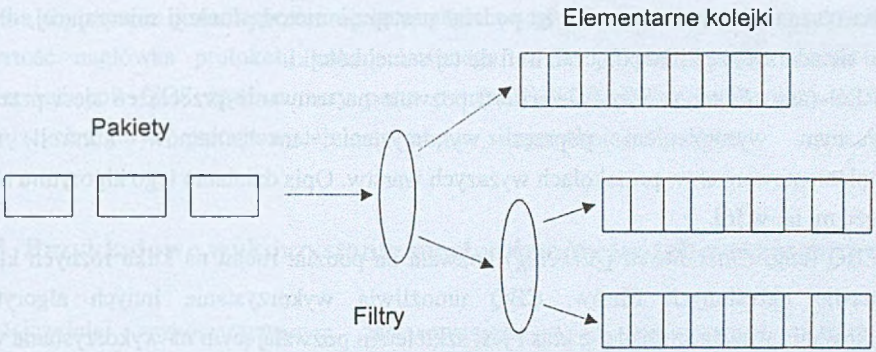
2. Metody kształtowania ruchu w Linuxie

2.1. Mechanizmy limitowania ruchu

Rutery wykorzystujące Linuxa oferują bardzo szerokie możliwości kształtowania ruchu. Od wersji 2.2 został całkowicie przebudowana implementacja stosu protokołów TCP/IP. Dzięki temu wykorzystując komputer pracujący w systemie Linux jako ruter można m.in. regulować pasmo (ang. *bandwith*) przydzielone poszczególnym komputerom, sprawiedliwie dzielić dostępne pasmo oraz równoważyć obciążenie serwerów. Parametry można regulować bazując na adresie fizycznym MAC, adresie IP, numerze portu, czasie dnia lub typ usługi. W łatwy sposób można także skonfigurować tunel tworzący wirtualną sieć prywatną (VPN) w sieci Internet [2] lub translacje adresów (NAT).

Konfiguracji parametrów trasowania dokonuje się za pomocą komendy *ip* będącej częścią pakietu *iproute2* autorstwa Aleksieja Kuzniecowa. Pakiet ten jest standardowo instalowany w niektórych dystrybucjach Linuxa (np. Redhat 7.2) oraz dostępny poprzez ftp m. in. z [4].

Kontrola ruchu odbywa się poprzez wybór algorytmu kolejkowania oraz określenie filtrów. Pakiety na podstawie nałożonego filtra są kierowane do odpowiedniej klasy. W ramach jednej klasy może następować podział na podstawie kolejnych filtrów na kolejne klasy ruchu, bądź pakiety zostają skierowane do elementarnej kolejki. Następnie wewnątrz niej następuje decyzja, czy wysłać je dalej, czy odrzucić. Metody filtrowania i kolejkowania pakietów mogą więc być zorganizowane w sposób hierarchiczny. Ilustruje to rys. 1.



Rys. 1. Organizacja metod filtrowania i kolejkowania pakietów
Fig. 1. Packet filtering and queuing methods organization

2.2. Metody kolejkowania pakietów

Kształtowanie ruchu odbywa się poprzez wybór odpowiedniej metody kolejkowania. Domyślna i najprostsza jest *pfifo_fast* będąca kolejką FIFO z kilkoma priorytetowymi klasami ruchu. Liczba klas może być zmieniona przez administratora, domyślnie jest to 3, a wybór odpowiedniej klasy następuje na podstawie pola TOS pakietu IP. Do kolejki z największym priorytetem jest kierowany ruch interaktywny, więc takie pakiety zostają odrzucone z najmniejszym prawdopodobieństwem. Użytkownik może także zmieniać długość kolejki. Istnieje także kolejka *bfifo* będąca prostą kolejką FIFO bez podziału na klasy priorytetowe, ale z możliwością zbierania statystyk przechodzącego ruchu.

Bardzo użytecznym algorytmem kolejkowania jest cieknące wiadro (ang. *Token Bucket Filter*). Umożliwia ono administratorowi ograniczenie dostępnego pasma do wyznaczonego poziomu. W algorytmie tym tworzony jest bufor (wiadro, ang. *bucket*) o określonej wielkości zawierający żetony generowane ze stałym natężeniem będącym parametrem sterowanym przez administratora. Wypuszczenie żetonu wiąże się z wypuszczeniem z kolejki określonej porcji danych. Jeżeli dane nadchodzą wolniej niż żetony, bufor się zapełnia. Poprzez określenie natężenia generowania żetonów regulujemy dostępne pasmo. Algorytm ten umożliwia także na buforowanie chwilowych przeciążeń dzięki określonej wielkości bufora z żetonami.

Stochastyczna sprawiedliwa kolejka SFQ (ang. *Stochastic Fairness Queuing*) pozwala na równe dzielenie pasma pomiędzy wiele połączeń. Jako połączenie jest rozumiana wymiana danych o takich samych adresach nadawcy i odbiorcy oraz protokole. Ruch jest dzielony na wiele kolejek FIFO, które następnie są opróżniane wedle algorytmu *round Robin*. Słowo

stochastyczna w nazwie oznacza, że podział następuje metodą funkcji mieszającej, dlatego może się zdarzyć, że kilka połączeń trafi do tej samej kolejki.

RED (ang. *Random Early Detection*) pozwala na usuwanie przeciążeń sieci przed ich faktycznym wystąpieniem poprzez wykorzystanie mechanizmów kontroli ruchu zaimplementowanych w protokołach wyższych warstw. Opis działania tego algorytmu można znaleźć m. in. w [6].

CBQ (ang. *Class Based Queueing*) pozwala na podział ruchu na kilka różnych klas na podstawie określonych filtrów. CBQ umożliwia wykorzystanie innych algorytmów kolejkowania wewnątrz każdej z klas i jest szkieletem pozwalającym na wykorzystanie wyżej opisanych metod filtrując ruch i dzieląc go na kilka różnych kolejek.

Metoda WRR (ang. *Weighted Round Robin*) rozdziela pasmo na kilka klas ruchu, wysyłając na przemian pakiety każdej z nich. Każdej z klas można przypisać wagę. Podział na klasy może następować na podstawie adresu IP lub MAC.

Kolejną interesującą dyscypliną kolejkowania jest *Dsmark*. Została opracowana dla sieci pracujących w standardzie DiffServ [3] pozwalającym na rozszerzenie sieci IP o zasady QoS. W sieciach DiffServ występuje podział na rutery wewnętrzne oraz zewnętrzne. Te drugie korzystając z pola TOS nagłówka IP oznaczają pakiety nadchodzące od klienta zgodnie z podpisaną z nim umową dzieląc je na kilka klas ruchu o różnej jakości usług. Wewnątrz sieci decyzja o traktowaniu pakietów odbywa się już tylko na podstawie pola TOS.

Dzięki dostępności kodu źródłowego systemu Linux można w łatwy sposób zaimplementować kolejne algorytmy sterowania ruchem. Opisane powyżej są najczęściej używanymi dostępnymi w ramach standardowej instalacji pakietu *iproute*, lecz jest także dostępnych wiele innych [2].

2.3. Filtrowanie pakietów

Klasyfikacji pakietu można dokonywać na podstawie jednego z trzech filtrów: *route*, *fw* oraz *u32*. Pierwszy z nich jest związany z tablicą routingu i może przypisywać pakiety do klas ruchu związanych z określoną trasą. Można go wykorzystać np. do ograniczenia ruchu dla wybranego komputera (adresu IP) lub ich grupy.

Filtr *fw* działa w połączeniu z programową ścianą ogniową działającą na ruterze. Firewall oznacza każdy datagram liczbą 32-bitową, na podstawie której jest on kierowany do określonej kolejki. W ten sposób można zrealizować filtrowanie pakietów również na podstawie informacji z protokołów wyższych niż TCP oraz UDP. Uruchomienie ściany ogniowej jednak dodatkowo obciąża ruter i może powodować zmniejszenie wydajności.

Najbardziej rozbudowany jest filtr *u32*. Umożliwia porównywanie nagłówków IP, TCP lub UDP z podanym wzorcem o wielkości bajtu, słowa lub długiego słowa. Porównanie

odbywa się poprzez podanie przesunięcia względem nagłówka i wzorca. Można także badać zawartość nagłówka protokołu enkapsulowanego w IP. W ten sposób można m.in. analizować pole TOS bądź numery portu transmisji TCP. Dzięki wykorzystaniu tablic haszujących algorytm pracuje wydajnie nawet przy silnym obciążeniu rutera.

3. Przykładowe wykorzystanie mechanizmów kształtowania ruchu

Najczęściej wykorzystywanym zastosowaniem wyżej opisywanych metod jest ograniczenie pasma dla określonego typu ruchu. Konfigurując filtr *u32* możemy wyodrębnić ruch skierowany do lub od wybranego komputera. Następnie można skierować go do kolejki TBF i ograniczyć pasmo dostępne dla wybranego użytkownika. Używając tej samej metody kolejkowania można ograniczyć ruch generowany przez wybrany protokół, np. FTP, lub wyodrębnić wybraną usługę i zagwarantować dla niej część pasma, np. dla wideokonferencji lub wideo na żądanie.

Istnieje także możliwość uzależnienia stosowanych filtrów od czasu. Jest to metoda użyteczna, gdy opłaty za łącze z Internetem są uzależnione od ilości przesyłanych danych – ograniczając pasmo dostępne w nocy obniżamy koszty firmy nie obniżając szybkości pracy sieci w ciągu dnia.

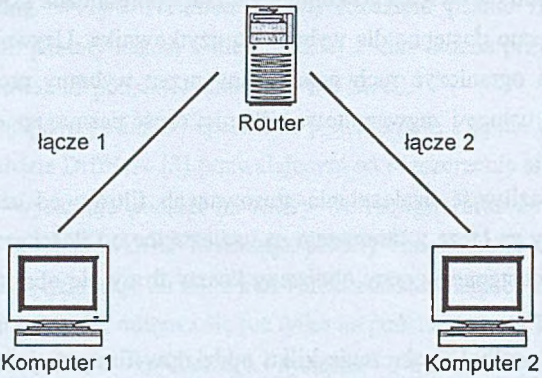
Pakiet *iproute* umożliwia połączenie kilku oddziałów firmy w jedną sieć korporacyjną (ekstranet) poprzez tunele w sieci Internet. Jednocześnie można ograniczyć pasmo dostępne dla ruchu „na zewnątrz", nie ograniczając ruchu wewnętrznego. Poprzez zarezerwowanie części pasma dla usługi VoIP można ograniczyć koszty rozmów telefonicznych wykorzystując do tego oprogramowanie do transmisji głosu.

Kolejka *bfifo* jest użyteczna dla dostawców Internetu pobierających opłaty uzależnione od ilości transmitowanych danych. Kierując ruch od każdego z klientów do osobnej kolejki otrzymamy liczbę bajtów przesłaną przez każdego z nich. Inną użyteczną tutaj metodą może być WRR – jeżeli mamy np. łącze dzielone przez kilku klientów, z których jeden chce płacić więcej, lecz wymaga za to udostępnienia większego pasma, możemy przydzielić mu większą wagę i proporcjonalnie podzielić koszty używania łącza. Nie wykorzystana część pasma będzie automatycznie dostępna dla pozostałych użytkowników tego samego łącza.

Poprzez odpowiednią konfigurację filtrów można także zabezpieczyć się przed niektórymi atakami na sieć komputerową. Aby zablokować ataki typu DoS, należy filtrować pakiety ICMP i ograniczyć dostępne dla nich pasmo.

4. Testy wybranych metod kolejkowania

W celu zapoznania się z możliwościami kształtowania ruchu w Linuxie zrealizowaliśmy w IITiS PAN testową instalację sieciową zbudowaną z 3 komputerów PC pracujących w systemie Linux RedHat 7.2, z których jeden pracował jako ruter, a pozostałe jako źródła i odbiorniki danych (komputer 1 i 2). W warstwie fizycznej skorzystano z sieci Ethernet na łączu 1 oraz sieci FastEthernet na łączu 2. Schemat połączenia komputerów przedstawia rys. 2. Ograniczona wielkość instalacji została podyktowana ograniczonymi zasobami sprzętowymi. Badana sieć nie była połączona z siecią lokalną instytutu, aby wyłączyć wpływ zewnętrznego ruchu na wyniki pomiarów



Rys. 2. Schemat połączenia testowej instalacji sieciowej
Fig. 2. Schema of network test bed installation

Badania zachłannej transmisji danych realizowano poprzez przesyłanie pliku protokołem FTP. Podczas niektórych testów także sprawdzano wpływ generowanego ruchu na szybkość pracy interaktywnej (telnet).

W ramach testów sprawdzono możliwość ograniczenia dostępnego pasma za pomocą metody kolejkowania TBF. Po ustawieniu szybkości transmisji na 100 kbit przesyłano plik o rozmiarze 10 MB. Zmierzona szybkość przesyłania pliku była mniejsza od zadanej wartości o kilka procent, ponieważ zadane pasmo obejmuje narzut wywołany przez nagłówki TCP/IP. W kolejnym eksperymencie do powyższego ograniczenia dodano kolejkę SFQ. Dzięki jej użyciu pasmo jest równomiernie dzielone pomiędzy wszystkie rodzaje ruchu. Zauważono duże przyśpieszenie pracy wykonywanej przy użyciu programu telnet, co wcześniej wiązało się z zauważalnymi opóźnieniami przy przesłaniu każdego znaku.

Opisane testy miały na celu wykazanie poprawności konfiguracji środowiska testowego. W przyszłości planujemy zebranie wyników pomiarowych porównujących zachowanie się

metod kolejkowania pakietów dla różnych rodzajów usług, do których wykorzystuje się sieć komputerową.

5. Podsumowanie

Ze względu na niskie koszty rutery zbudowane ze zwykłych komputerów wykorzystujących system Linux stają się bardzo popularne. Poprzez ich odpowiednie skonfigurowanie można znacząco poprawić jakość usług oferowanych ich użytkownikowi. Opisane w rozdziale 3 zastosowania opisanych metod mogą usprawnić wykorzystanie dostępnego pasma lub taryfikację transmisji danych.

W ramach dalszych prac zostaną wykonane pomiary działania wszystkich opisanych w rozdziale 2 metod kształtowania ruchu w celu opracowania zestawienia porównującego cechy i zastosowania tych metod. Zamierzamy również dołączyć do testowanej sieci komputerowej serwer usługi wideo na żądanie, aby zbadać wpływ wyboru rodzaju mechanizmów kształtowania ruchu na jakość transmisji multimedialnej.

LITERATURA

1. McDysan D.: QoS & Traffic Management in IP & ATM Networks. McGraw-Hill 2000.
2. Maxwell G., van Mook R. van Oosterhout M., Schroeder P. Spaans J.: Linux Advanced Routing & Traffic Control HOWTO, http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html
3. RFC 2475 - An Architecture for Differentiated Services, http://www.ietf.org/rfc/rfc2475.txt?number=2475
4. Kuzniecow A.: IP Route. ftp://ftp.icm.edu.pl/pub/Linux/iproute
5. Krawczyk P.: Sterowanie przepływem danych w Linuxie 2.2, http://ceti.pl/~kravietz/cbq/.
6. Nowak S.: Wykorzystanie mechanizmu RED w regulacji ruchu sieciowego, ZN Pol. Śl. Studia Informatica Vol. 22, No 3(45), Gliwice 2001

Recenzent: Dr inż. Mirosław Skrzewski

Wpłynęło do Redakcji 17 kwietnia 2002 r.

Abstract

The article presents the traffic shaping functions build in Linux. Beginning from version 2.2 of the system kernel the IP protocol stack has been completely rewritten and enhanced with *iproute2* software. New implementation allows to filter incoming and outgoing traffic and give different treatment to the different type of network traffic. Incoming packets are divided into traffic classes, which can be organized hierarchically. User can select different queuing algorithm for every traffic class.

The work presents a few most popular queuing disciplines. The available filters are also presented. The third chapter contains some possible usages of traffic management functions, what should bring to the reader to the more practical view. The authors have build small testing network to become acquainted to the technology. Some measurements made there are described in the fourth chapter.