

1. Ogólna charakterystyka ćwiczenia

W dzisiejszych czasach przy rosnącej liczbie przesyłanych danych, głosu, wideo dąży się do tego, by zapewnić odpowiednią jakość przekazu przez sieć IP. Przesyłanie głosu, wideo i danych w jednej sieci IP nie jest proste, ponieważ różne rodzaje ruchu wymagają zapewnienia różnych poziomów usług. Dane wideo wymagają dużej przepustowości i spójnego opóźnienia. Nie wymaga się przy tym małego opóźnienia lecz opóźnienie powinno być zgodne by zapobiec niepożądanym efektom niepewności przesyłu. Przesyłanie ruchu głosowego nie wymaga wysokiej przepustowości ale w tym przypadku powinno zapewnić się bardzo krótkie opóźnienia sieciowe. Nadmierne opóźnienie sesji głosowej wprowadza niepożądane efekty np. echo które jest bardzo trudne do wyeliminowania. Interaktywna sesja Telnet, narzędzie pozwalające na zdalną pracę na odległych komputerach, wymaga niskich opóźnień. Transfer plików w sieci korzysta z dużej przepustowości, przy czym jest najbardziej wyrozumiały ze wszystkich typów ruchu sieciowego, tolerując umiarkowane opóźnienia

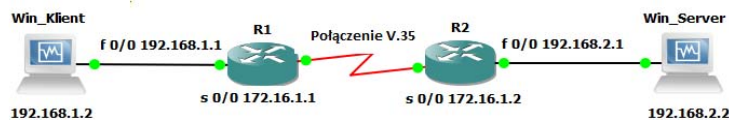
Aby zapewnić te wszystkie usługi w jednej sieci ip, należy użyć różnych mechanizmów QoS, które to warunkują strumienie ruchu sieci na podstawie podanych wymagań. W niniejszym ćwiczeniu zostaną zbadane różnego rodzaju mechanizmy Quality of Service. Klasyfikowanie oraz oznaczanie pakietów ruchu wychodzącego przez router brzegowy jest bardzo istotną rzeczą do poprawnego nadawania priorytetów danego ruchu. W niniejszym ćwiczeniu będzie do wykonania kilka różnych testów, które pozwolą bardziej zrozumieć działanie sieci w której występują mechanizmy QoS.

Celem niniejszego ćwiczenia jest zbadanie oraz skonfigurowanie wybranych mechanizmów zapewnienia parametrów jakościowych w systemach sieciowych.

2. Plan wykonywania ćwiczenia laboratoryjnego

Zadanie 1. Ograniczenie przepustowości metodą Shaping dla ruchu z odpowiednim polem DSCP

W niniejszym badaniu należy sprawdzić działanie ograniczenia przepustowości, dzięki różnieniu ruchu sieciowego z odpowiednim polem DSCP. Ograniczenie przepustowości będzie miało miejsce dzięki statycznej metodzie *Shaping*. W zadaniu będą wykorzystane dwie stacje, jedna będzie pełniła rolę klienta druga zaś serwera. Stacja klienta będzie połączona z routerem brzegowym R1, a stacja serwer będzie połączona z routerem R2. Połączenie między routerami będzie realizowane łączem szeregowym o ograniczonej przepustowości do 1 Mb/s. Wymagana konfiguracja będzie realizowana na routerze brzegowym R1. Do wykonania zadania niezbędny będzie program iperf3, który będzie uruchomiony na obu komputerach w różnej konfiguracji. Do analizy wykresów tego zadania może posłużyć środowisko wireshark z funkcją IO Graphs albo program DU Meter. Do wykonania zadania układ sieciowy będzie wyglądał następująco:



Rys 3.1 Układ sieciowy do badania mechanizmów QoS

- Do wykonania zadania należy utworzyć trzy klasy. Każda z klas ma być dopasowana do różnego pola DSCP. Poniżej przykładowa tabela zawierająca klasy z dopasowanymi polami DSCP.

Tab 1 Przykładowe rozróżnienie klas dzięki polu DSCP

Nazwa klasy	Przypisane pole DSCP
High	Af43
Average	Af22
Low	Af11

- Następnie należy utworzyć zasadę, do której dopasować odpowiednie klasy utworzone w punkcie wcześniejszym. Dla każdej z klas ograniczyć prędkość transmisji dzięki statycznej metodzie *Shaping*

Tab 2 Przykładowe klasy z przypisanymi ograniczeniami prędkości

Nazwa klasy	Przypisane pole DSCP	Ograniczona prędkość
High	Af43	600 kb/s
Average	Af22	300 kb/s
Low	Af11	100 kb/s

- Przypisanie utworzonej zasady do interfejsu *serial 0/0* w kierunku wychodzącym

Po wykonaniu powyższej konfiguracji należy uruchomić program iperf3 na obu komputerach. Na komputerze klienta uruchomić program realizujący zadanie wysyłania ruchu. Na stacji serwera uruchomić go w trybie nasłuchiwania. Następnie należy wykonać testy przesłania ruchu z odpowiednim polem DSCP. Wyniki należy zarejestrować poprzez wybrany przez studentów program.

Zadanie 2. Zbadanie mechanizmu kolejkowania Custom Queuing.

W kolejnym zadaniu należy sprawdzić działanie mechanizmu kolejkowania CQ, które będzie polegało na ograniczeniu przepustowości dla trzech różnych typów ruchu. W tym wypadku należy skorzystać z trzech sposobów emulowania ruchu. Ruch protokołu UDP należy przesłać za pomocą programu iperf3 z komputera klienta na serwer. Drugim typem ruchu będzie pobieranie pliku testowego *testftp.zip* poprzez protokół FTP z komputera klienta na komputer serwera. Trzecią transmisją będzie pobieranie pliku *testhttp.zip* protokołu HTTP przez komputer klienta na komputer serwera.

- Do wykonania zadania należy utworzyć trzy listy dostępowe poleceniem. Pierwsza z list powinna umożliwiać ruch protokołu *udp* z komputera klienta na komputer serwera. Druga z list powinna umożliwiać ruch *ftp-data* z komputera klienta na komputer serwera. Ostatnia z list dostępowych powinna umożliwiać ruch protokołu http w takim samym kierunku jak dwie pozostałe listy..

- Następnie należy utworzyć pierwszą kolejkę typu CQ, dla protokołu ip i przypisać odpowiednią listę dostępową ruchu protokołu *udp*. Następnie należy daną kolejkę ograniczyć przepustowością do 8000 bajtów. Te same kroki trzeba uczynić dla kolejnych dwóch typów ruchu. Druga kolejka powinna ograniczać przepustowość do 4000 bajtów. Trzecia kolejka powinna ograniczać przepustowość do poziomu 3000 bajtów. Należy także utworzyć kolejkę domyślną, która będzie ograniczała ruch niedopasowany do żadnych ACL na prędkość 1000 bajtów. Łącze szeregowe pomiędzy komputerami ograniczyć do 128 kb/s.

Tab 3 Przykładowe listy dostępowe z odpowiednim typem ruchu sieciowego wraz z ograniczeniami prędkości

Lista dostępową	Typ ruchu	Ograniczona prędkość(bajty)
101	Udp	8000
102	ftp-data	4000
103	http	3000
Default	niedopasowany	1000

- Na koniec należy przypisać utworzoną kolejkę typu CQ do interfejsu *serial 0/0*

Po wykonaniu powyższej konfiguracji należy uruchomić program *iperf3* na obu komputerach. Na komputerze klienta uruchomić program realizujący zadanie wysyłania ruchu. Na stacji serwera uruchomić go w trybie nasłuchiwanie. Wysyłanie ruchu protokołu *udp* przez program *iperf3* powinno trwać co najmniej 60 sekund. Czas ten jest tak określony, by móc w tym samym czasie uruchomić pobieranie obu plików testowych, dla dwóch pozostałych typów ruchu sieciowego. Wyniki należy zarejestrować na wykresie IO Graphs programu Wireshark z odpowiednimi filtrami.

Zadanie 3. Zbadanie mechanizmu kolejkowania Weighted Fair Queuing.

W kolejnym zadaniu należy sprawdzić działanie mechanizmu kolejkowania WFQ. Zadanie będzie polegało na wysyłaniu trzech transmisji przy wyłączonym mechanizmie WFQ oraz przy jego włączeniu. Tak jak we wcześniejszych zadaniach jednym z ruchów będzie pobieranie pliku *test_ftp.zip* za pomocą protokołu FTP. Drugi z ruchów będzie polegał na pobraniu pliku *test_http.zip* poprzez protokół http. Trzecim rodzajem ruchu będzie to ruch protokołu TCP z przypisanym krytycznym polem DSCP o wartości ToS 184. Przepustowość łącza w tym badaniu powinna wynosić 128 kb/s.

- W pierwszej kolejności należy wyłączyć mechanizm WFQ na interfejsie *serial 0/0*. Dzieje się to dlatego iż, mechanizm ten jest domyślnie włączony na routerach o łączach 2 Mbps lub o niższej przepustowości.
- Po wykonaniu zadania wyłączenia mechanizmu WFQ, należy przeprowadzić test trzech rodzajów ruchu wysyłając wszystkie jednocześnie. Wysyłanie ruchu protokołu *udp* przez program *iperf3* powinno trwać co najmniej 60 sekund. Czas ten jest tak określony, by móc w tym samym czasie uruchomić pobieranie obu plików testowych, dla dwóch pozostałych typów ruchu sieciowego. Zarejestrować wyniki na wykresie IO Graphs z zastosowaniem odpowiednich filtrów programu Wireshark.
- Następnie należy przejść do interfejsu *serial 0/0* i włączyć mechanizm WFQ.
- Po wykonaniu czynności włączenia mechanizmu WFQ, należy ponownie przeprowadzić test trzech rodzajów ruchu wysyłając wszystkie jednocześnie. Zarejestrować wyniki na wykresie IO Graphs z zastosowaniem odpowiednich filtrów programu Wireshark.

Zadanie 4. Zbadanie mechanizmu TBF na ograniczonej przepustowości za pomocą mechanizmu Shaping.

W zadaniu czwartym niniejszej instrukcji należy zbadać działanie mechanizmu Token Bucket Filter. Zadanie to będzie polegało na porównaniu kształtowania *Shaping average* oraz kształtowania *Shaping peak* z wykorzystaniem mechanizmu TBF. Algorytm TBF posiada trzy parametry, które będą określane w zasadzie kształtowania. Parametry te to CIR, Bc, oraz Be. CIR określa, ile danych może być przesłanych lub przekazanych średnio w czasie. Parametr Bc odpowiada temu, ile żetonów zostanie pobranych z wiadra w danym okresie czasu. Parametr Be określa się po to, by móc powiększyć dane wiadro z żetonami o wartość Be. Oba parametry Bc oraz Be są po to, aby mieć większą kontrolę nad parametrem CIR. Wzór na działanie algorytmu to $CIR = \frac{Bc}{Tc}$. W celu realizacji zadania będzie trzeba wykonać poniższe kroki:

- Należy utworzyć dwie klasy. Jedna z klas będzie reprezentowała kształtowanie typu *shape average*, druga kształtowanie *shape peak*.
- Następnie utworzyć należy zasadę i przypisać do niej pierwszą z klas. W zasadzie należy ustawić parametr CIR = 32000 b/s. Parametr Bc ustawić na takim poziomie, aby zauważyć działanie pobierania żetonów z interwałem równym 7 sekund. Parametr Be w tym wypadku równy 0.
- Następnie przypisać do interfejsu *serial 0/0* utworzoną zasadę.
- Przepustowość na łączu szeregowym powinna wynosić 512 kb/s
- Po wykonaniu powyższej konfiguracji sprawdzić działanie poprzez pobranie pliku *test_fip.zip* z komputera klienta na komputer serwera. Wyniki zarejestrować dzięki IO Graphs z odpowiednim filtrem.

Różnicą pomiędzy mechanizmem *shape average* a *shape peak* polega na tym, że w pierwszym przypadku mechanizm w odpowiednim czasie wysłę tylko określoną ilość żetonów nie przekraczając tej wartości. W *shape peak* sytuacja ta wygląda inaczej. Wartość maksymalna to suma parametrów Bc oraz Be, a wzór wygląda następująco $PIR = Bc + Be$. W wypadku równości oby parametrów wzór $PIR = CIR * (1 + \frac{Be}{Bc})$ przyjmuje następującą formę

$PIR = 2 * CIR$. Taką sytuację należy przedstawić w kolejnym punkcie zadania.

- Po przebadaniu pierwszego kształtowania *Shape average*, utworzyć należy zasadę i przypisać do niej drugą z klas. W zasadzie należy ustawić parametr CIR = 32000 b/s.

Parametr Bc ustawić na takim poziomie, by móc zauważyć działanie pobierania żetonów z interwałem równym 7 sekund. Parametr Be w tym wypadku będzie równy parametrowi Bc.

- Po wykonaniu powyższej konfiguracji sprawdzić działanie poprzez pobranie pliku *test_fip.zip* z komputera klienta na komputer serwera. Wyniki zarejestrować dzięki IO Graphs z odpowiednim filtrem.

W sprawozdaniu należy zamieścić opis sposobu konfiguracji poszczególnych mechanizmów QoS. Wykresy utworzone w poszczególnych ćwiczeniach oraz wnioski nabyte po wykonaniu zadań powinny być zamieszczone w sprawozdaniu.

3. Wymagania BHP

Zgodnie z podanymi na pierwszych zajęciach i potwierdzonymi przez studentów zasadami obowiązującymi w pomieszczeniu, w którym odbywają się ćwiczenia. Stosowny regulamin BHP jest też wywieszony w pomieszczeniu laboratorium.

4. Literatura

[1] Kevin Dooley & Ian J. Brown, *Cisco receptury*, Helion, Gliwice, 2004.

[2] Tabela wartości pól DSCP & TOS:
<https://www.tucny.com/Home/dscp-tos>

[3] Tutorial programu iperf:
<https://openmaniak.com/iperf.php>

[4] Konfiguracja algorytmu Token Bucket Filter:
http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfdts.html

[5] Konfiguracja Custom Queueing:
http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfq.html

[6] Konfiguracja Weighted Fair Queueing:
http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfwfq.html

[7] Kształtowanie metodą Policing oraz Shaping:
http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfpoli.html