

Telekomunikacja



Akademia Górniczo-Hutnicza w Krakowie
- Katedra Telekomunikacji -

SYMULACJA USŁUG QoS W ŚRODOWISKU OPNET



**Dobrzyniecki Konrad
Gruszczyk Piotr**

Praca przygotowana na potrzeby przedmiotu „Sieci Teleinformatyczne” prowadzonego przez mgr inż. Jacka Rzęsę na IV roku Telekomunikacji.

Kraków, grudzień 2005r.

Spis treści

1.	Wprowadzenie	3
2.	Omówienie QoS.....	3
2.1	Wstęp.....	3
2.2	Model oparty na rezerwacji zasobów	4
2.3	Model oparty na rozróżnianiu klas ruchu.....	4
2.4	Klasyfikacja pakietów	4
2.5	Klasyfikacja i etykietowanie CAR , PBR, BGP Policy Propagation	6
2.6	Zatory i zarządzanie nimi.....	6
3.	Opis symulatora Opnet.....	11
4.	Przeprowadzone symulacje.....	12
4.1	Założenia projektowe sieci.....	12
4.2	Symulacja sieci bez zaimplementowanego QoS	13
4.3	Symulacja sieci z zaimplementowaną kolejką FIFO.....	14
4.4	Symulacja sieci z zaimplementowanym PQ – Priority Queuing.....	15
4.5	Symulacja sieci z zaimplementowanym CQ – Custom Queuing.....	16
4.6	Symulacja sieci z zaimplementowanym CQ with LLQ.....	17
4.7	Symulacja sieci z zaimplementowanym WFQ.....	18
4.8	Symulacja sieci z zaimplementowanym WFQ with LLQ.....	19
5.	Podsumowanie.....	20
6.	Bibliografia.....	20

1. Wprowadzenie

Od samego początku powstania sieci komputerowej pojawił się problem odpowiedniego zagwarantowania jakości usług przesyłanych przez nią. Problem ten zyskał szczególną uwagę w dzisiejszym stanie rozwoju teleinformatyki. Usługi głosowe, wideo, telewizja interaktywna, telefonia internetowa, są niezwykle wrażliwe na jakość ich przesyłania. Brak gwarancji jakości dla takich usług jest równoważne z zaprzeczeniem ich głównych założeń.

W niniejszej pracy zawarliśmy wprowadzenie do tematu: „Jakość usług (QoS)”, krótką wzmiankę o środowisku Opnet, w którym przeprowadzaliśmy symulacje oraz wreszcie same symulacje zaimplementowania usług QoS w sieci komputerowej.

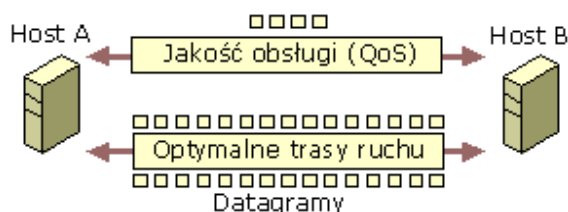
Niniejsza praca przygotowana przez Konrada Dobrzynieckiego i Piotra Gruszczyka, studentów IV roku Telekomunikacji Akademii Górniczo-Hutniczej w Krakowie, na potrzeby przedmiotu „Sieci Teleinformatyczne” prowadzonego przez mgr inż. Jacka Rzęsę.

2. Omówienie QoS

2.1 Wstęp

Jakość usługi (QoS, Quality of Service) to zestaw wymagań, które musi spełnić sieć, aby zapewnić odpowiedni poziom usług transmisji danych. Jest to zbiór technologii, które pozwalają użytkownikom otrzymywać od sieci przewidywalny poziom usług, patrząc z punktu przepustowości, opóźnienia oraz zmienności tego opóźnienia.

Gwarancja QoS oznacza taki poziom usług, który pozwala programowi przesyłać dane z określoną szybkością i dostarczać je w określonym przedziale czasu.



Rys 1. Jakość obsługi [5]

Jeżeli chcemy, aby strumień danych otrzymał wymaganą przez nas jakość usług, każdy węzeł w sieci musi być poinformowany o tych wymaganiach. Możemy zrobić to w dwojaki sposób, poprzez:

- etykietowanie pakietów - pakiet niesie ze sobą informacje o wymaganiach,

- sygnalizację – zaimplementowany jest specjalny protokół do powiadamiania.

Stąd też pojawiły się dwa modele QoS różniące się sposobem informowania sieci:

- model oparty na rezerwowaniu zasobów,
- model oparty na rozróżnianiu klas ruchu.

2.2 Model oparty na rezerwacji zasobów

W tym modelu został zaimplementowany protokół sygnalizacyjny i poprzez niego aplikacje zgłaszają sieci swoje wymagania przepustowości łącza, maksymalnego opóźnienia pakietów. Węzeł sieci po otrzymaniu takiej informacji z protokołu sprawdza dostępność zasobów, a następnie w przypadku stwierdzenia, że istnieją odpowiednie zasoby dla transportu informacji, rezerwuje je. Jest to dobra forma zapewnienia jakości usługi, pod warunkiem, że ruch generowany przez aplikację nie przekracza ustalonej wcześniej wielkości. Ruch ten jest ciągle sprawdzany przez węzeł sieciowy, w przypadku jego zwiększenia zostaje ograniczony do poziomu zgłoszonych wcześniej parametrów.

Przykładem takiego modelu jest IETF IntServ z użyciem protokołu RSVP.

2.3 Model oparty na rozróżnianiu klas ruchu

Model ten charakteryzuje się brakiem jawnego zgłaszania przez aplikację wymagań dotyczących jakości usługi, jak to było poprzednio. Tym razem strumień danych generowany przez aplikację zaliczany jest do jednej z klas ruchu. Klasa taka oznaczona jest odpowiednią etykietą zawartą w pakietach danych. W tym przypadku węzły sieci nie dostarczają konkretnej bezwzględnej jakości usług (np. konkretna wartość zarezerwowanego pasma), lecz względną – priorytet przesyłania danych zależy od klasy.

Przykładem takiego modelu jest IETF DiffServ.

2.4 Klasyfikacja pakietów

Klasyfikacja pakietów ma na celu zidentyfikowanie pakietów pod względem ich wymagania, co do jakości usług. Może odbywać się to za pomocą:

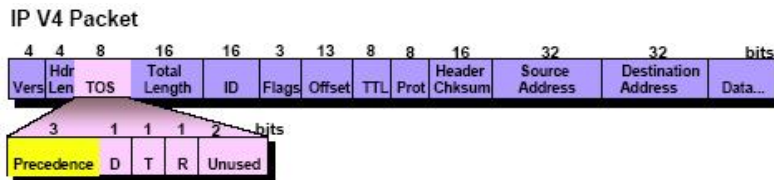
- źródłowego/docelowego adresu IP,
- interfejsu,
- numeru portów TCP/UDP,
- adresów MAC.

Innym sposobem klasyfikacji pakietów są też mechanizmy warstwy aplikacji (NBAR – *Network-Based Application Recognition*).

Pakiety po klasyfikacji oznaczane są etykietą pozwalającą na szybką identyfikację klasy usług pakietu przez węzły sieci. Technikami etykietowania pakietów są:

- a) warstwa 3 modelu OSI/ISO:
 - IPv4 IP Precedence Field,

- IPv4 DiffServ Differentiated Services Field,
 - IPv6 DiffServ Differentiated Services Field,
- b) warstwa 2 modelu OSI/ISO:
- MPLS Ex/Cos Field,
 - 802.1d (802.1p+q) User Priority Field,
 - ISL User Priority Field.



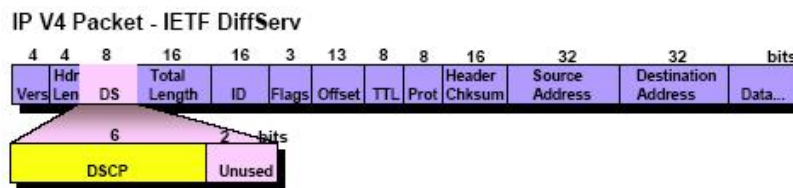
TOS - Type of Service Field
 - Definicja w RFC 791.
Precedence
 - 3-bitowe pole oznaczające klasę usług
DTR Bits
 - Flagi oznaczające priorytet pod względem opóźnienia, przepustowości i niezawodności

Rys 2. IP Precedence [3]

Znaczenie bitów jest następujące:

- Bity 0-2: Priorytet
- Bit 3: 0 = Normalne Opóźnienie, 1 = Małe Opóźnienie
- Bit 4: 0 = Normalne Przejście, 1 = Przyspieszone Przejście
- Bit 5: 0 = Normalna Niezawodność, 1 = Podwyższona Niezawodność
- Bity 6-7: Zarezerwowane do wykorzystania w przyszłości

Wartość priorytetu rośnie, gdy rośnie wartość wyznaczona przez bity 0-2.



DS - Differentiated Services Field
 - Definicja w RFC2747
DSCP - Differentiated Services Codepoint Field
 - 6-bitowe pole oznaczające jakość usług
 - 32 wartości zestandaryzowane, pozostałe do zastosowań lokalnych
 - Wartości zestandaryzowane zapewniają wsteczną kompatybilność z IP Precedence

Rys 3. DSCP [3]

2.5 Klasyfikacja i etykietowanie CAR , PBR, BGP Policy Propagation

CAR (Committed Access Rate) jest to mechanizm umożliwiający policing, klasyfikację oraz etykietowanie. Klasyfikacja pakietów odbywa się na podstawie adresów IP, numerów portów TCP/UDP, interfejsów wejściowych, adresów MAC oraz IP precedence. Dla pakietów które spełniają kryteria klasyfikacyjne mierzone jest zajmowane przez nie pasmo. Istnieje możliwość zmiany wartości pola IP precedence, jeśli zajmowane pasmo przekracza ustaloną wartość.

PBR (Policy-Based Routing) to mechanizm, który umożliwia definiowanie polityki routingu, a także klasyfikowanie i etykietowanie pakietów. Klasyfikacja pakietów odbywa się w podobny sposób jak w CAR. Pakiety, które spełniają kryteria kwalifikacyjne mają ustawiane pole IP precedence na zadaną wartość.

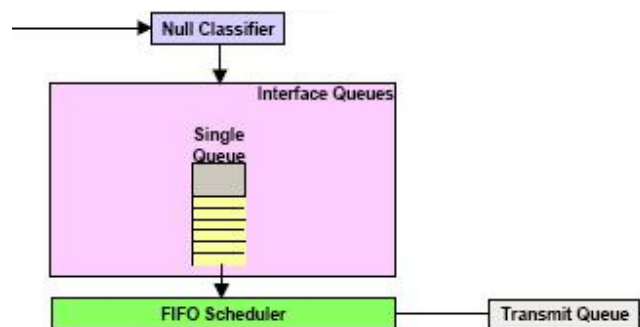
BGP Policy Propagation jest wykorzystywane do propagacji polityki klasyfikacji dla ruchu zwrotnego. Klasyfikacja pakietów w tym mechanizmie jest oparta na: access list, community list i autonomous system paths. Na podstawie informacji BP router tworzy tablicę zawierającą adresy docelowe oraz odpowiadające im wartości IP precedence.

2.6 Zatory i zarządzanie nimi

Zator na węźle sieciowym ma miejsce, gdy prędkość napływu pakietów do interesu wyjściowego przekracza jego przepustowość. Mówiąc o zarządzaniu zatorami mamy na myśli zapewnienie odpowiedniej jakości usług aplikacjom, których działanie zależą od opóźnienia –

- zapewnienie im priorytetu w przesyłaniu danych. Wyjściem z sytuacji jest kolejkowanie pakietów i ustalenie kolejności, w jakiej mają one zostać wytransportowane. Trzy główne zadania, jakie stoją przed modelem zarządzania zatorami to:
- klasyfikacja – według zadanych kryteriów,
- kolejkowanie – umieszczenie w kolejce na podstawie klasyfikacji,
- planowanie – ustalenie kolejności transmisji.

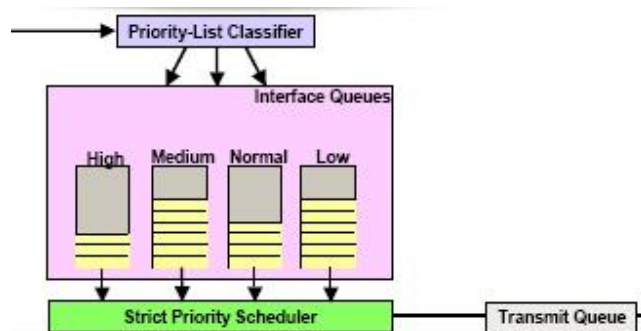
FIFO Queuing



Rys 3. FIFO Queueing [3]

Klasyfikacja	- nie ma,
Kolejkowanie	- istnieje pojedyncza kolejka, pakiety umieszczane są w takiej kolejności, w jakiej napłynęły,
Planowanie	- transmisja pakietów na zasadzie: “pierwszy przyszedł pierwszy wyszedł”, - bardzo mały narzut czasowy na planowanie, - obsługa aplikacji krytycznych lub wrażliwych na opóźnienia jest niedostateczna, - dobre, proste zastosowanie dla interfejsów niezatłoczonych.

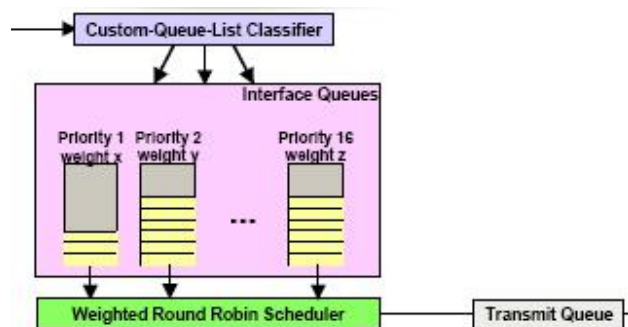
PQ – Priority Queuing



Rys 4. Priority Queuing [3]

Klasyfikacja	- klasyfikacja odbywa się na bazie protokołów warstwy 3, interfejsu wejściowego, access-list oraz rozmiaru pakietu,
Kolejkowanie	- istnieją cztery kolejki, w których są umieszczane nadchodzące pakiety, umieszczanie wg priorytetu,
Planowanie	- obsługa kolejek według bezwzględnych priorytetów, - kolejka o najwyższym priorytecie jest obsługiwana tak długo, aż znajdują się w niej pakiety, - kolejka o niższym priorytecie czeka, aż ta o wyższym zostanie całkowicie wypróżniona, - kolejki o najniższych priorytetach przy dużych ruchach bardzo często są całkowicie niedopuszczone do medium.

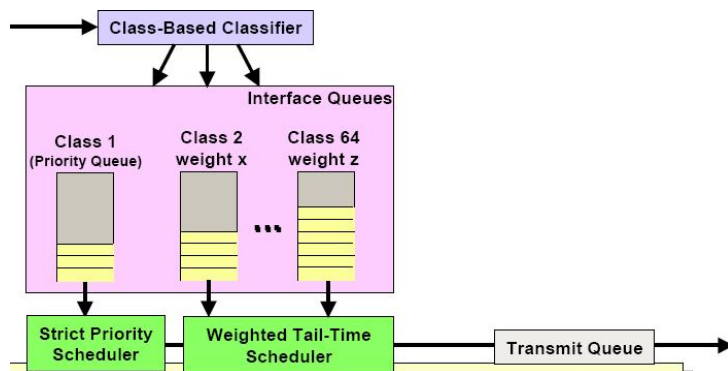
CQ – Custom Queuing



Rys 5. Custom Queuing [3]

Klasyfikacja	- klasyfikacja odbywa się na bazie protokołów warstwy 3, interfejsu wejściowego, access-list oraz rozmiaru pakietu,
Kolejkowanie	- pakiety na podstawie klasyfikacji umieszczane są w jednej z 16 kolejek (17 to kolejka systemowa),
Planowanie	- obsługiwanie kolejek odbywa się cyklicznie (round robin), - dla każdej kolejki mamy określoną maksymalną wartość bajtów, jakie możemy wytransportować, - gwarancja pasma dla każdej kolejki.

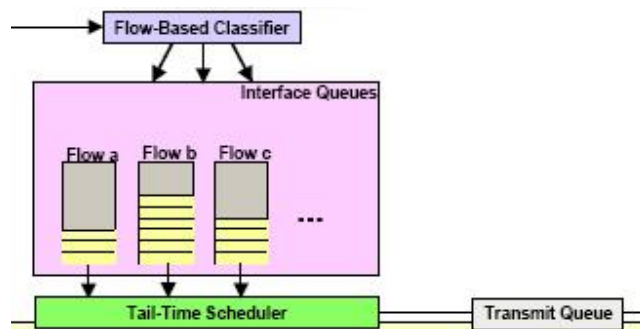
CQ with LLQ– Custom Queuing with Low Latency Queuing



Rys 6. Custom Queuing with Low Latency Queuing[3]

Klasyfikacja	<ul style="list-style-type: none"> - podział na klasy na podstawie interfejsu wejściowego, protokołów i access listy, - pakiety, które nie pasują do żadnej reguły są przypisywane do klasy “class-default”,
Kolejkowanie	<ul style="list-style-type: none"> - pakiety na podstawie klasyfikacji umieszczane w jednej kolejce priorytetowej lub w jednej z 64 kolejek odpowiadających klasom,
Planowanie	<ul style="list-style-type: none"> - bezwzględne pierwszeństwo transmisji mają pakiety z kolejki priorytetowej, - jednak jej pasmo nie powoduje „zagłodzenia” innych kolejek.

FQ - Flow-Based Fair Queuing



Rys 7. Flow-Based fair Queuing [3]

Klasyfikacja	<ul style="list-style-type: none"> - automatyczny podział na “konwersacje” na podstawie adresów źródłowych i docelowych oraz numerów portów TCP i UD,
Kolejkowanie	<ul style="list-style-type: none"> - pakiety na podstawie klasyfikacji kierowane są do osobnych kolejek dla każdej konwersacji,
Planowanie	<ul style="list-style-type: none"> - emulowany jest system TDM polegający na transmitowaniu z każdej kolejki w jednym cyklu jednego bitu, - każdy pakiet oznaczony jest czasem zakończenia transmisji w wirtualnym systemie TDM, - do transmisji wybierany jest pakiet z najkrótszym czasem zakończenia, - konwersacje o małym zapotrzebowaniu na pasmo dostają go tyle ile potrzebują, a konwersacje o dużym zapotrzebowaniu dzielą resztę pasma równo pomiędzy siebie, - aplikacje krytyczne lub wrażliwe na opóźnienie nie uzyskują żadnego priorytetu.

WFQ – Flow-Based Weighted Fair Queuing

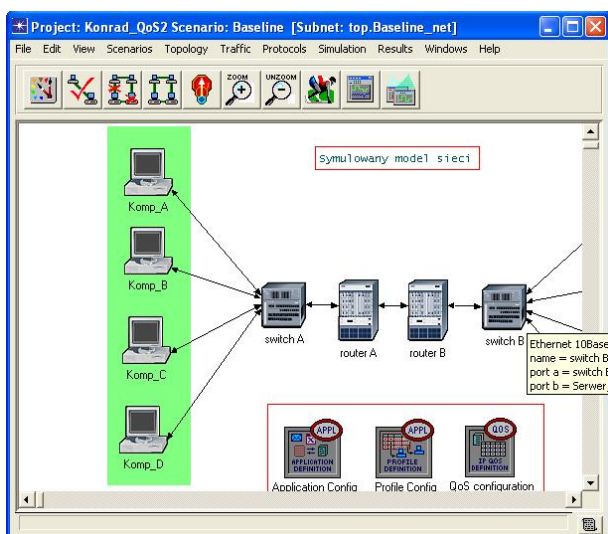
Klasyfikacja	- tak jak w FQ,
Kolejkowanie	- tak jak w FQ,
Planowanie	<ul style="list-style-type: none">- dla każdego pakietu wyliczana jest waga zależna od wartości IP precedence,- waga uwzględniana jest w wirtualnym systemie TDM na potrzeby wyznaczenia czasu zakończenia transmisji,- pierwszy będzie wytransportowany pakiet z najmniejszym czasem zakończenia,- wagi poszczególnych pakietów pozwalają na bardzo dobre traktowanie aplikacji krytycznych.

3. Opis symulatora Opnet

Oprogramowanie OPNET IT Guru [9] jest wirtualnym środowiskiem służącym do modelowania, symulowania i analizy pracy systemów informatycznych, z uwzględnieniem ich infrastruktury obejmującej technologie, protokoły, urządzenia i aplikacje. OPNET umożliwia rozwiązywanie problemów związanych z wydajnością aplikacji, testowaniem zmian w konfiguracji serwerów i routerów oraz planowaniem rozwoju sieci. Symulacje przeprowadzone w wirtualnym środowisku sieciowym pozwalają przewidzieć zachowanie



Rys 8. Oprogramowanie Opnet IT Guru



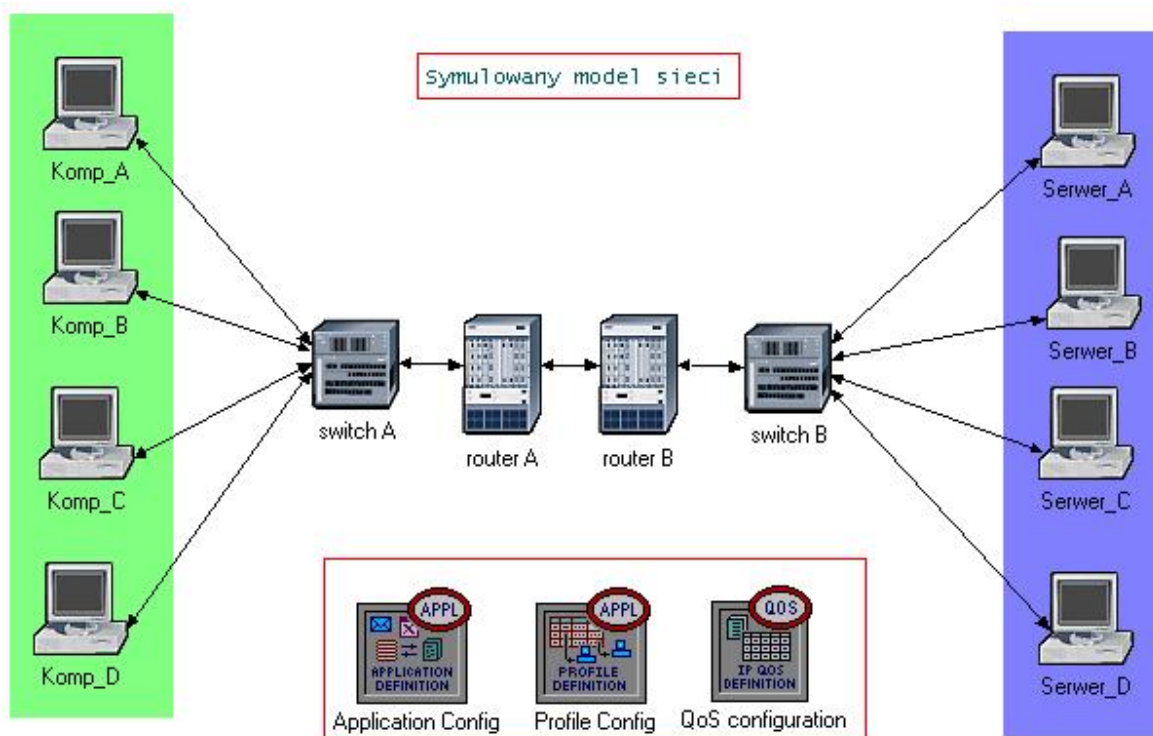
Rys 9. Widok okna głównego Opnet.

infrastruktury sieciowej w różnych sytuacjach bez eksperymentowania na prawdziwej sieci.

Przedstawicielem producenta środowiska Opnet na Polskę jest firma Passus z Warszawy [8]. Nie promując kryptoreklamy musimy zaznaczyć, że pomoc techniczną ma świetną – na e-mail z zapytaniem o problem pojawiający się w oprogramowaniu podczas symulacji naszej sieci odpowiedź dostaliśmy niezwłocznie.

4. Przeprowadzone symulacje

4.1 Założenia projektowe sieci



Rys 10. Symulowany model sieci.

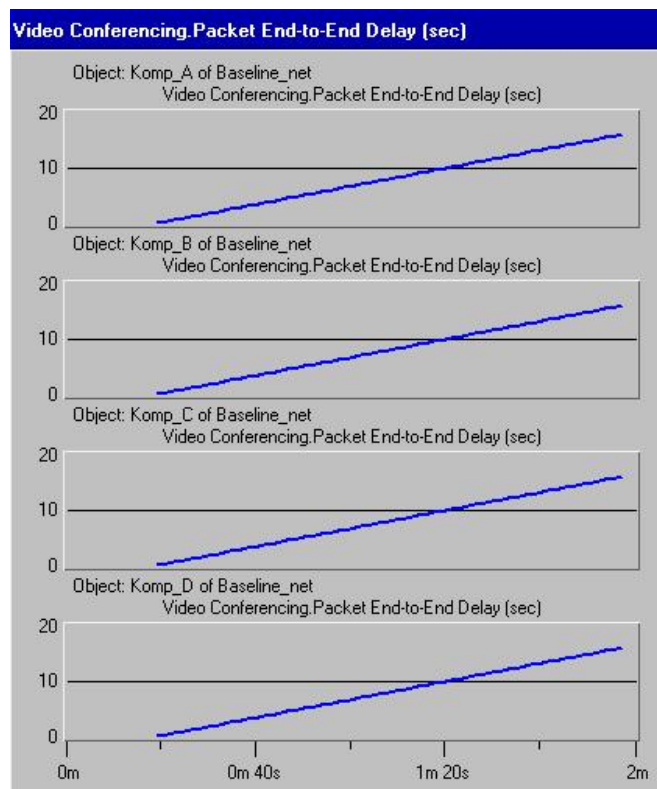
Sieć użyta podczas symulacji składała się ze stacji roboczych będących klientami serwerów wideo. Każdy z nich używał innej wartości TOS (Type Of Service) podczas transportu danych. I tak:

- Komputer_A – (background, ToS=1),
- Komputer_B – (standard, ToS=2),
- Komputer_C – (excellent, ToS=3),
- Komputer_D – (streaming, ToS=4).

Wąskim gardłem dla tej sieci było łącze pomiędzy dwoma routerami. W routerze A zastosowaliśmy mechanizm kolejkowania.

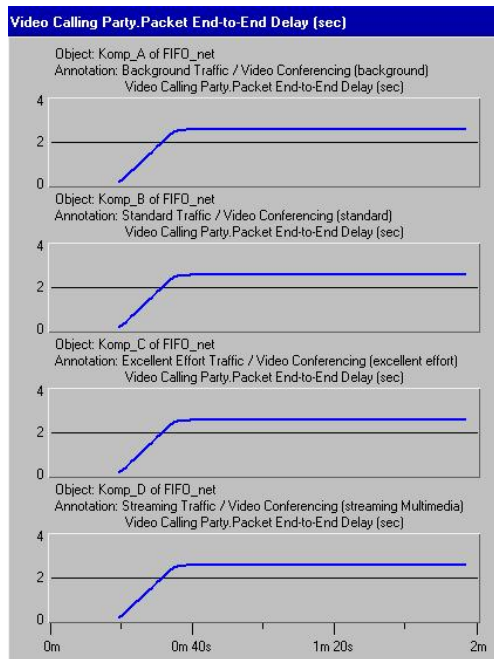
4.2 Symulacja sieci bez zaimplementowanego QoS.

Przypadek ten pokazuje nam, że od czasu rozpoczęcia ruchu przez stacje robocze (ustawione na 10 sekund) mamy jednostajny wzrost opóźnienia spowodowany brakiem jakichkolwiek gwarancji usług zaimplementowanych na serwerze. Na poziomej osi mamy pokazany czas symulacji, pionowa ukazuje opóźnienie End-To-End pakietów wideo.



Rys 11. Opóźnienie dla każdego z ruchów.

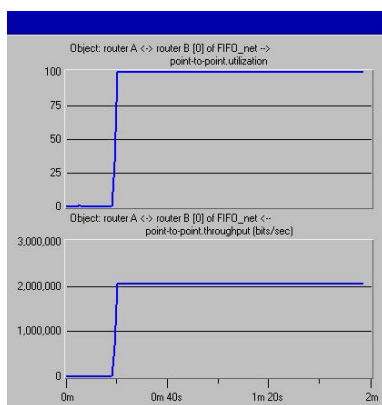
4.3 Symulacja sieci z zaimplementowaną kolejką FIFO.



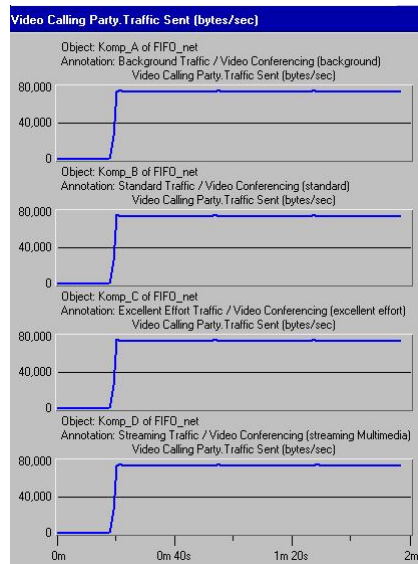
Rys 12. Opóźnienie dla każdego z ruchów.

Zapewnienie jakości ruchu polega na kolejkowaniu ruchu w routerze A według mechanizmu „FIFO”. Nie mamy w tym przypadku klasyfikacji, router nie zważa na pola ToS. Przychodzący ruch wrzucany jest do kolejki według zasady: „Pierwszy Przyszedł, Pierwszy Wychodzi” – „First In First Out”. Wszystkie przychodzące pakiety traktowane są w równorzędny sposób, nie mamy w tym przypadku gwarancji usług.

Na wykresie zauważamy początkowo małe opóźnienia – kolejka wyjściowa nie zdążyła się jeszcze zappełnić. Potem mamy wzrost opóźnienia aż do stałej wartości 3 sekund. W czasie, gdy kolejka jest zappełniona, przychodzące do routera pakiety są przez niego odrzucane.

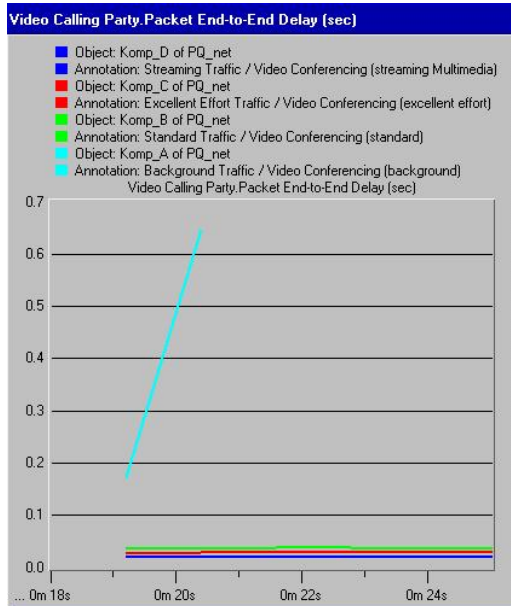


Rys 13. Wykorzystanie kanału i przepływność na łączu Router A-Router B.

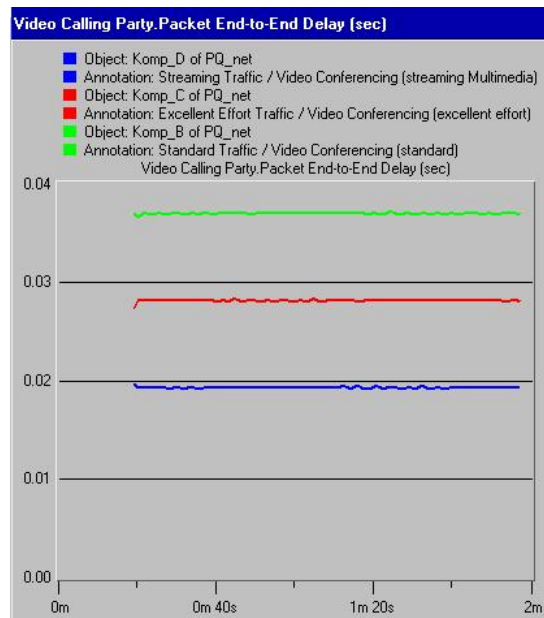


Rys 14. Dane wysłane z Komp A-D.

4.4 Symulacja sieci z zaimplementowanym PQ – Priority Queuing

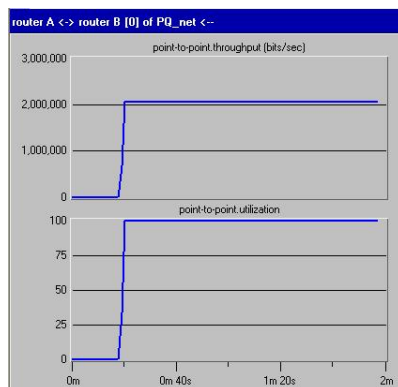


Rys 15. Opóźnienie dla każdego z ruchów..

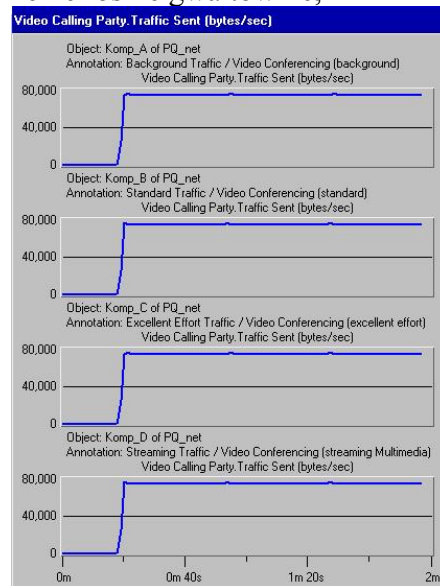


Rys 16. Powiększenie dla Komp B-D.

Zapewnienie jakości ruchu polega tym razem na kolejkowaniu ruchu w routerze A według mechanizmu „PQ Priority Queuing”. Klasyfikacja odbywa się na bazie ToS, im wyższy numer tym ważniejszy pakiet. Jako pierwsze transportowane są pakiety z Komp_D (streaming – ToS=4). Potem, jeśli nie ma już w niej nic kolejno transportujemy z Komp_C (excellent-ToS=3), Komp_B (standard, ToS=2), Komp_A (background, ToS=1). Zauważamy w miarę dobre parametry opóźnień dla Komp_D, C, B, dla Komp_A z powodu zbyt małej przepływności łącza RouterA-RouterB opóźnienie rośnie gwałtownie, uniemożliwiając praktycznie przepływ ruchu Komp_A - Serwer_A. Wnioskujemy, że metoda ta daje bardzo dobre wyniki, ale tylko dla najwyższych wartości TOS, pakiety z najmniejszymi ToS są w tych warunkach prawie ignorowane.

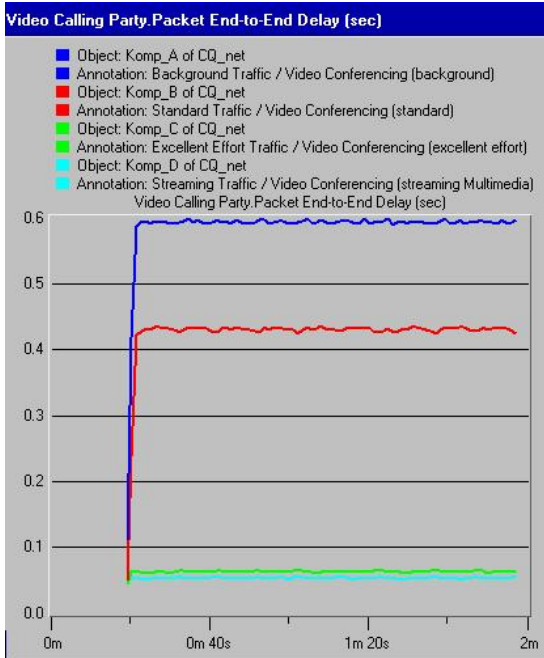


Rys 17. Wykorzystanie kanału i przepływność na łączu Router A-Router B.

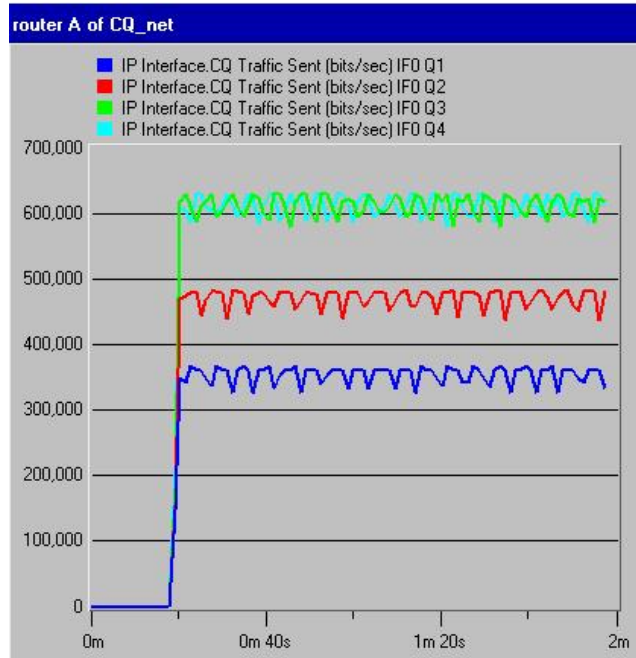


Rys 18. Dane wysłane z Komp A-D.

4.5 Symulacja sieci z zaimplementowanym CQ – Custom Queuing

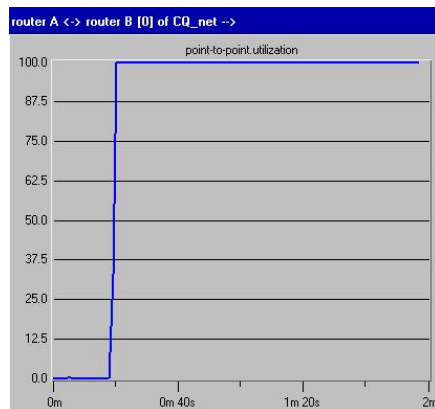


Rys 19. Opóźnienie dla każdego z ruchów..

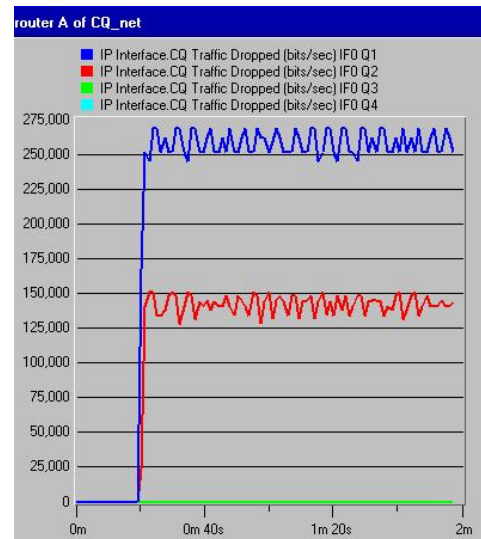


Rys 20. Ruch wysyłany na interfejsach routera A.

Tym razem symulujemy Custom Queuing. Zasada tutaj jest taka, że z każdej z kolejek wejściowych routera do kolejki wyjściowej wybierane są pakiety, z tym że różna jest ich ilość w zależności od priorytetu kolejki. I tak, w naszej symulacji mamy: Komp_D (ToS=4) – jego pakiety trafiają do kolejki, gdzie za jednym cyklem wybieranych jest 10000B, Komp_C (ToS=3) – 8000B, Komp_B (ToS=2) – 6000B, Komp_A (ToS=1)–4000B. Parametr ten ustawiamy w *QoS Configuration->Custom Queuing Profiles->row->Byte Count*. Patrząc na wykresy zauważamy, że 8000B na cykl w zupełności wystarczy dla przepływu pakietów. Jeśli nie ma informacji w kolejce, obsługiwana jest od razu następna (utilization 100%). Mamy także w tym przypadku gwarancję usług dla wszystkich.

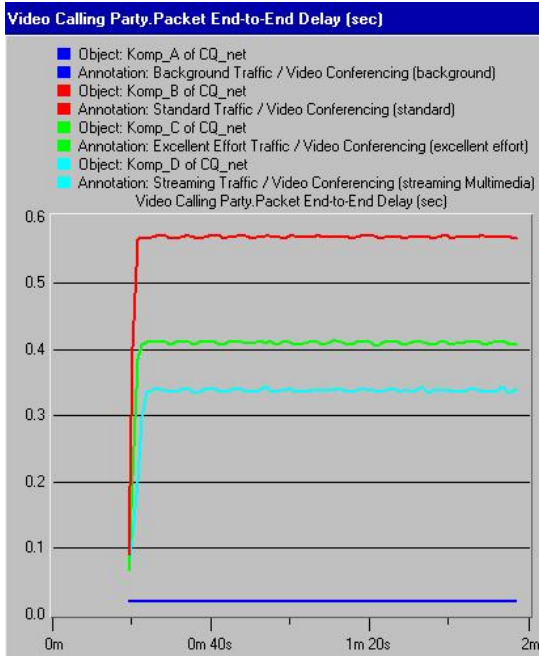


Rys 21. Wykorzystanie kanału i przepływność na łączu Router A-Router B.

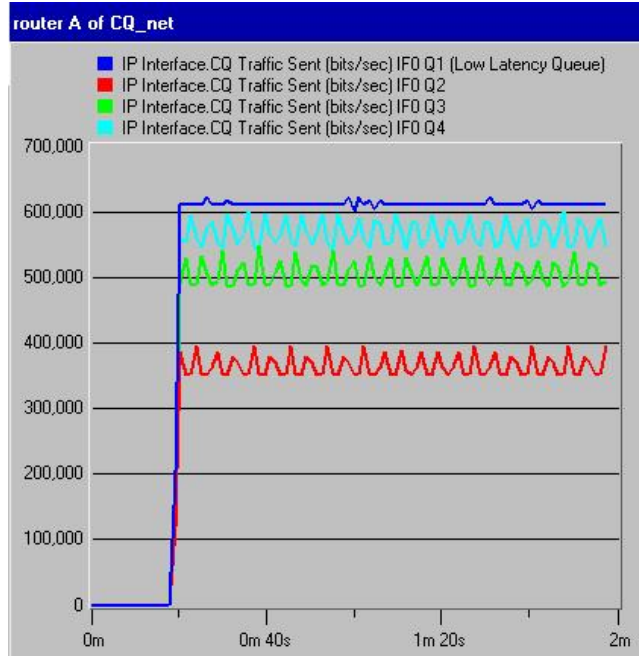


Rys 22. Ruch odrzucany na routerze A.

4.6 Symulacja sieci z zaimplementowanym CQ with LLQ



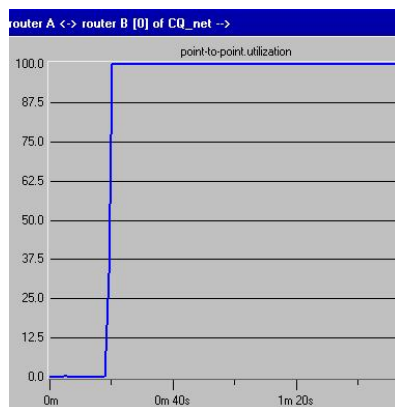
Rys 23. Opóźnienie dla każdego z ruchów..



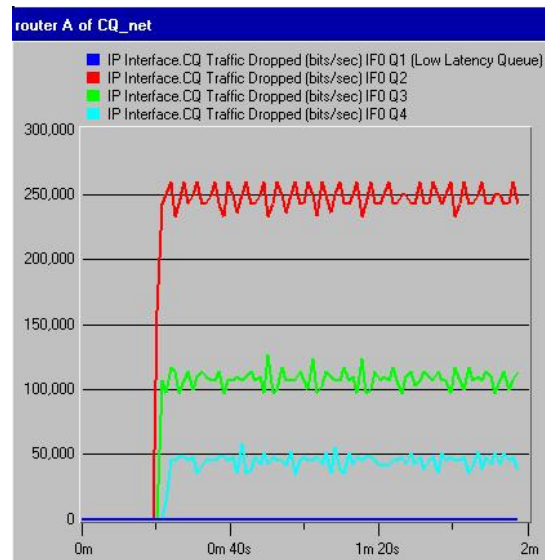
Rys 24. Ruch wysyłany na interfejsach routera A.

Kolejny krok to symulacja Custom Queuing with Low Latency Queue. W tym przypadku mamy jedną kolejkę uprzywilejowaną - Low Latency Queuing – pozwalającą w tym przypadku na ruch bez opóźnień. Tylko wtedy, gdy ta kolejka jest pusta istnieje możliwość transportowania pakietów z innych kolejek zgodnie z mechanizmem CQ omówionym na poprzedniej stronie. Parametr ten ustawiamy w *QoS Configuration->Custom Queuing Profiles->row->Queue Category->Low Latency Queuing*.

Teorię potwierdza symulacja – kolejka z Komp_A (niebieska), nie ma żadnych opóźnień, a pozostałe 3 zostają klasyfikowane zgodnie z ToS.

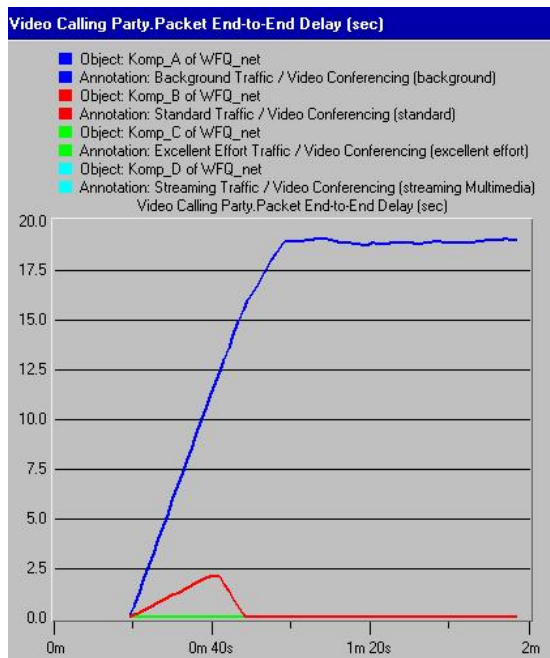


Rys 25. Wykorzystanie kanału na łączu Router A-Router B.

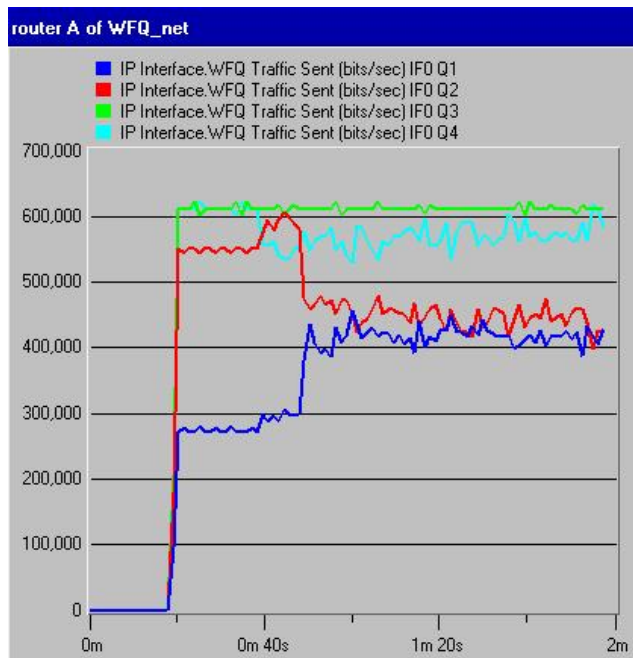


Rys 26. Ruch odrzucany na routerze A.

4.7 Symulacja sieci z zaimplementowanym WFQ



Rys 9. Opóźnienie dla każdego z ruchów..

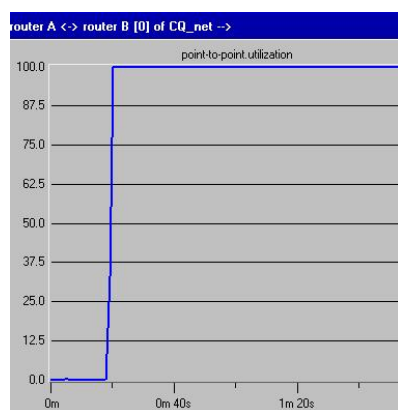


Rys 10. Powiększenie dla Komp B-D.

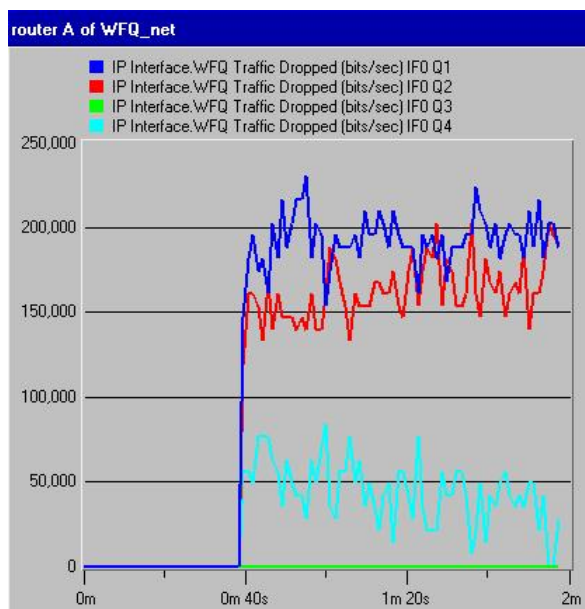
W tej symulacji mamy do czynienia z implementacją WFQ. Tym razem ruch wysyłany jest do interfejsu wyjściowego licząc wagę każdego pakietu, biorąc pod uwagę takie czynniki jak adres nadawcy, odbiorcy, ToS, a także portów TCP/UDP.

Parametr ten ustawiamy w *QoS Configuration->WFQ Profiles->Queues Configuration->row->Weight*.

Widzimy, że dla Komp_C wyliczona waga była najlepsza, mamy dla niego najmniejsze opóźnienie, waga dla Komp_A była najgorsza wprowadzając 20 sekundowe opóźnienie pakietów.

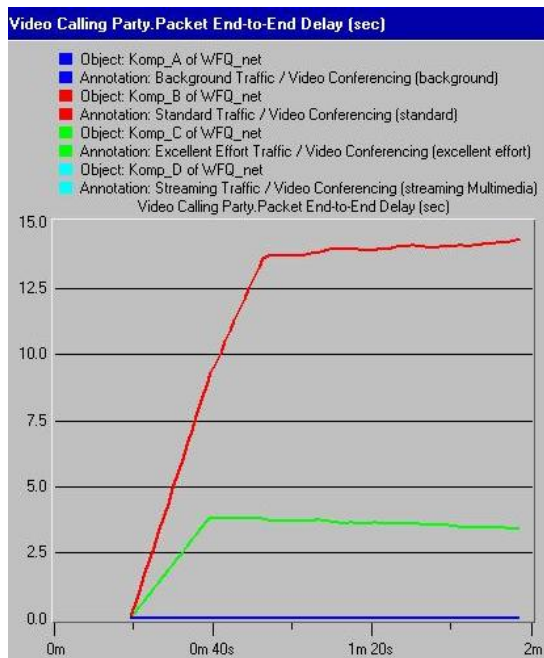


Rys 10. Wykorzystanie kanału i przepływność na łączu Router A-Router B.

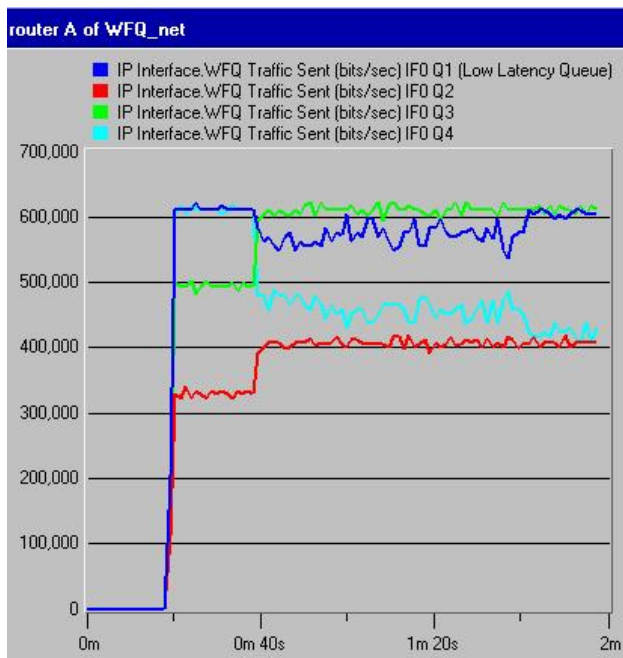


Rys 10. Dane wysłane z Komp A-D.

4.7 Symulacja sieci z zaimplementowanym WFQ with LLQ

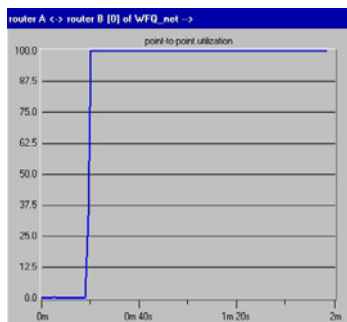


Rys 9. Opóźnienie dla każdego z ruchów..

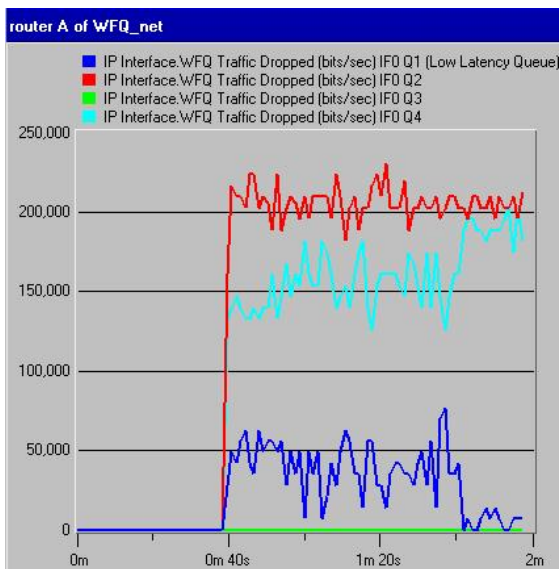


Rys 10. Powiększenie dla Komp B-D.

Podobna sytuacja jak powyżej, z tym, że mamy dodaną kolejkę o priorytecie najwyższym, niezależnym od innych dla Komp_A – pełna gwarancja zerowego opóźnienia pakietów.



Rys 10. Wykorzystanie kanału i przepływność na łączu Router A-Router B.



Rys 10. Dane wysłane z Komp A-D.

5. Podsumowanie

Przedstawiony wstęp teoretyczny, a wreszcie przeprowadzone symulacje dały nam obraz tego, jak ważny jest problem dostarczania jakości podczas przesyłania danych w sieci. Stojąc na progu czasów, gdy usługi multimedialne, i to te z najwyższej półki – telefonia, telewizja interaktywna, staną się niedługo standardem, możemy być pewni, że zagadnienia QoS w najbliższym czasie nie stracą na znaczeniu, a wręcz przeciwnie staną się jedną z najważniejszych gałęzi rozwoju usług teleinformatycznych.

Mamy nadzieję, że czytelnik znajdzie w naszej pracy proste i jasne wytłumaczenie podstaw QoS. Autorom opracowania projekt ten dostarczył dużej satysfakcji z poznanej wiedzy, zarówno teoretycznej, jak i obsługi środowiska symulacyjnego Opnet.

6. Bibliografia

- [1] Specification of Guaranteed Quality of Service – RFC 2212
<ftp://ftp.rfc-editor.org/in-notes/rfc2212.txt>
- [2] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm
- [3] <http://www.wcss.wroc.pl/cisco/>
- [4] <http://rainbow.mimuw.edu.pl/SO/Linux/Temat08/IP.html>
- [5] <http://www.microsoft.com/technet/prodtechnol/default.msp>
- [6] http://akson.sgh.waw.pl/%7Echopin/ios120/12cgr/qos_c/qcpart2/qcconman.htm
- [7] http://www.alcatel.pl/cmsimages/pl/QoS_tcm55-356871635.pdf
- [8] <http://www.passus.pl>
- [9] <http://www.opnet.com>