AKADEMIA GÓRNICZO-HUTNICZA
im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Katedra Telekomunikacji

AUTOREFERAT

mgr. inż. Jerzego Domżała

rozprawa doktorska nt. “Congestion Control in Flow-Aware Networks” (Sterowanie przeciążeniami w sieciach zorientowanych na przepływy (FAN))

Promotor: prof. dr hab. inż. Andrzej Jajszczyk

Kraków 2009

Geneza rozprawy doktorskiej

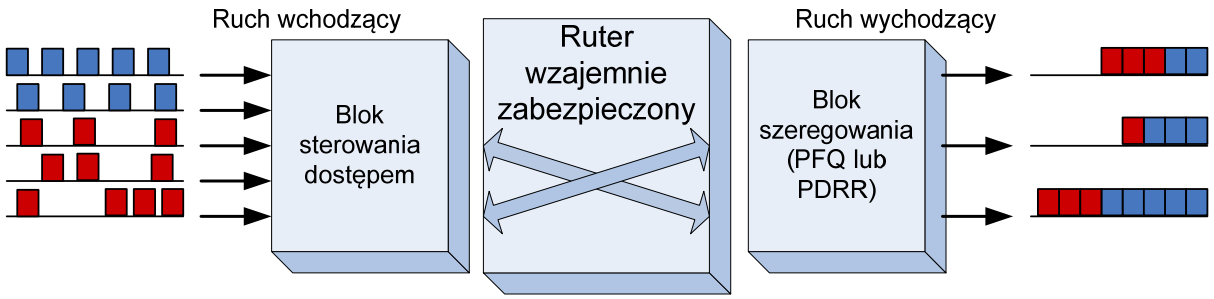
Istniejące sieci transmisji danych były projektowane głównie z nastawieniem na możliwość transmisji niegwarantowanej (*best effort*). Jednakże wraz z szybkim rozwojem usług sieciowych, na istniejące architektury nakładane są coraz większe wymagania. Usługi typu *Voice-over-IP* (VoIP) czy *Video-on-Demand* (VoD) wymagają nie tylko znacznej przepływności łączy, ale też małych opóźnień i niewielkich strat. Ruch generowany przez usługi tego typu jest więc nieelastyczny, z kolei np. przesyłanie danych przy pomocy protokołu FTP, które z powodzeniem można przeprowadzić w sieci realizującej transmisję niegwarantowaną jest przykładem usługi generującej ruch elastyczny. Aby usługi nieelastyczne mogły funkcjonować, konieczne jest stworzenie w sieci warunków gwarantujących odpowiednią jakość tychże usług. Są to tzw. gwarancje QoS (*Quality of Service*).

Stosowane obecnie rozwiązania (oparte w większości przypadków na architekturach *IntServ* i *DiffServ*) wymagają wprowadzenia specyficznych mechanizmów, takich jak rezerwacja pasma, kształtowanie ruchu, wykorzystywanie klas ruchowych, stosowanie odpowiedniej polityki zarządzania dostępną przepływnością. Mechanizmy te sprawdzają się dobrze w sieciach z jednym typem ruchu. W sytuacji, gdy naturalnym wydaje się traktowanie platformy IP jako wspólnej dla przesyłania ruchu elastycznego i nieelastycznego, mechanizmy te nie są zbyt korzystne. Dotyczy to przede wszystkim nieefektywnego wykorzystywania dostępnej przepustowości łączy, a co za tym idzie zbyt wysokich kosztów użytkowania sieci. Odpowiedzią na te niedogodności wydaje się być propozycja sieci zorientowanej na przepływy – FAN (*Flow-Aware Networking*). Architektura ta korzystając z mechanizmów sterowania dostępem do łącza (*admission control*) oraz szeregowania na poziomie przepływów (*per-flow scheduling*) pozwala na lepsze gospodarowanie dostępnymi przepływnościami. Zalety FAN uwidaczniają się przede wszystkim w sytuacji występowania „wąskiego gardła" w sieci, gdyż najczęściej zagadnienie zapewnienia pożądanej jakości obsługi zaczyna mieć znaczenie dopiero w chwili gdy łączny ruch oferowany przekracza przepustowość łączy.

Wychodząc naprzeciw wymaganiom QoS, w propozycji FAN założono występowanie dwóch typów ruchu: strumieniowego oraz elastycznego. Pierwszy z nich, przewidziany głównie do transmisji strumieni dźwiękowych bądź wizyjnych, wymaga przede wszystkim małych opóźnień, tolerując czasem niewielkie straty pakietów. Przepływy tego typu mają z reguły zmienną szybkość bitową, w zależności od zastosowanego kodowania. Drugi typ ruchu, przewidziany do transmisji danych, nie toleruje żadnych strat pakietów, jest za to bardzo tolerancyjny w kwestii opóźnień. Dla przepływów pierwszego typu stosuje się wysoki priorytet. Pozwala to na transmisję danych audio i wideo bez opóźnień, a także pozwala np. na bezproblemowe uruchamianie różnego rodzaju aplikacji wymagających transmisji w czasie rzeczywistym, jak na przykład gier sieciowych, czy aplikacji

związanych z głosem i ruchomymi obrazami. Drugi typ ruchu jest przewidziany przede wszystkim do transmisji danych. Przydzielona temu typowi ruchu przepływność łączy jest proporcjonalnie dzielona pomiędzy wszystkie przepływy.

Schemat blokowy rutera wzajemnie zabezpieczonego jest przedstawiony na Rys. 1.



Rys. 1. Architektura rutera wzajemnie zabezpieczonego

Sterowanie dostępem w każdym węźle sieci FAN zapewnia odpowiednie warunki dla istniejących przepływów w sytuacji wystąpienia natłoku. Tu podejmowane są decyzje o obsłużeniu lub odrzuceniu nowego przepływu. Wszystkie przepływy obsługiwane w danej chwili są rejestrowane na specjalnej liście chronionych przepływów PFL (*Protected Flow List).* Jeśli identyfikator przepływu napływających pakietów jest dodany do tej listy, to pakiety obsługiwane są natychmiastowo. Jeżeli jest inaczej, uruchamiany jest mechanizm sterowania dostępem. Jeśli łącze jest w stanie natłoku (transmisja nadchodzących pakietów nie jest możliwa), pakiety są odrzucane, co jest jednoznacznym sygnałem dla użytkownika, że łącze jest przeciążone. Gdy jednak łącze nie znajduje się w stanie natłoku, pakiety są obsługiwane natychmiastowo, a identyfikator przepływu jest dodawany do PFL. Pomiar zajętości łączy, w tym stwierdzenie czy łącze jest w stanie natłoku, jest dokonywany w sposób ciągły przez układ szeregujący (*scheduler*). Zapisy dokonywane w PFL nie są trwałe. Jeśli przez określony czas nie napływają pakiety o identyfikatorze danego przepływu, identyfikator ten jest usuwany z listy. Jest to rozwiązanie podobne do miękkiego stanu utrzymywanego przez protokół RSVP. Jednym z mechanizmów szeregowania pakietów w sieciach FAN jest *Priority Fair Queueing* (PFQ) [11,13,14]. Zakłada on wykorzystanie zmodyfikowanego algorytmu *Start-time Fair Queueing* (SFQ). Pakiety przepływów o szybkości mniejszej od tzw. szybkości sprawiedliwej (*fair rate*) są umieszczane w kolejce w taki sposób, by opuszczały ją w pierwszej kolejności. Działanie to zapewnia priorytetowe traktowanie przepływów strumieniowych o wystarczająco małej szybkości.

W bloku szeregowania rutera periodycznie mierzone są dwa parametry:

fair_rate – czyli chwilowa wartość szybkości dla danego łącza, która jest lub może być realizowana przez dowolny przepływ transmitujący dane przez łącze,

priority_load – czyli chwilowa wielkość ruchu z priorytetem, mierzona jako stosunek ruchu priorytetowego wpływającego do łącza do szybkości łącza.

W PFQ, *fair_rate* jest mierzony wg poniższego wzoru:

$$fair\_rate = \frac{\max\{S \times C, (vt(t_2) - vt(t_1)) \times 8\}}{t_2 - t_1} \quad (1)$$

gdzie vt(*t*) jest wartością parametru *virtual_time* (charakterystycznego dla algorytmu SFQ i reprezentującego początkowy czas nadawania fikcyjnego przepływu wysyłającego pakiety o długości 1 B pomiędzy pakietami innych przepływów, zgodnie z założeniami algorytmu) w chwili czasu *t*, $t_2 - t_1$ jest okresem czasu, mierzonym w sekundach, w którym jest dokonywany pomiar, *S* jest okresem czasu braku aktywności w wysyłaniu pakietów w obserwowanym przedziale czasu oraz *C* jest przepływnością łącza w bit/s.

Wartości parametru *priority_load* są obliczane z poniższego wzoru:

$$priority\_load = \frac{(pb(t_2) - pb(t_1)) \times 8}{C(t_2 - t_1)} \quad (2)$$

gdzie p*b*(*t*) jest wartością licznika zwiększaną o długość pakietu w bajtach, gdy pakiet przychodzi do rutera, ($t_2 - t_1$) jest okresem czasu, mierzonym w sekundach, w którym dokonywany jest pomiar oraz *C* jest przepływnością łącza w bit/s.

Wartości obu parametrów mają decydujące znaczenie w bloku sterowania dostępem do łącza, który zezwala na akceptację pakietu lub nakazuje go odrzucić. Gdy chwilowa wartość parametru *fair_rate* jest mniejsza niż minimalna dopuszczalna wartość tego parametru (*min_fair_rate*) lub chwilowa wartość parametru *priority_load* jest większa niż maksymalna dopuszczalna wartość tego parametru (*max_priority_load*) łącze znajduje się w stanie przeciążenia i pakiety przepływów dotychczas nie zaakceptowanych w bloku sterowania dostępem nie mogą być wysłane.

W implementacji rutera wzajemnie zabezpieczonego z algorytmem PFQ stosuje się kolejkę PIFO (*Push-In, Push-Out*). Pakiety przepływów strumieniowych, czyli takich których wielkość

zakolejkowanych danych nie przekracza wartości maksymalnej dopuszczalnej jednostki transmisyjnej MTU (*Maximum Transmission Unit*) są umieszczane na końcu kolejki i tym samym są obsługiwane wcześniej niż pakiety pozostałych przepływów (elastycznych). W ten sposób w ruterach FAN z algorytmem PFQ realizowane jest niejawne różnicowanie jakości obsługi.

Drugim z zaproponowanych mechanizmów szeregowania pakietów w sieciach FAN jest *Priority Deficit Round Robin* (PDRR), będący rozszerzeniem mechanizmu DRR (*Deficit Round Robin*) o funkcje priorytetyzowania ruchu [8, 9, 12]. Obsługa pakietów polega na kolejnym przeglądaniu kolejek przepływów transmitujących dane i pobieraniu do wysłania pakietów znajdujących się na czele kolejek. Przy spełnieniu odpowiednich warunków (ilość danych w kolejce jest mniejsza niż dopuszczalny kwant *Q*, ustawiony na wartość MTU), pakiety trafiają do wyjściowej kolejki priorytetowej. Mechanizmy sterowania dostępem do łącza oraz szeregowania są od siebie ściśle uzależnione.

W bloku szeregowania rutera wzajemnie zabezpieczonego z algorytmem PDRR periodycznie mierzone są te same dwa parametry co w przypadku algorytmu PFQ.

fair_rate jest mierzony wg poniższego wzoru:

$$fair\_rate = \frac{\max\{S \times C, FB \times 8\}}{t_2 - t_1} \tag{3}$$

gdzie *FB* jest liczbą bajtów przesłanych przez przepływy elastyczne w trakcie obserwowanego okresu czasu, $t_2 - t_1$ jest okresem czasu, mierzonym w sekundach, w którym dokonywany jest pomiar, *S* jest okresem czasu braku aktywności w wysyłaniu pakietów w obserwowanym przedziale czasu oraz *C* jest przepływnością łącza w bit/s.

Wartości parametru *priority_load* są obliczane ze wzoru (2).

W porównaniu do DRR, algorytm PDRR zmniejsza czasy opóźnień dla pakietów przepływów wysyłających dane z niską szybkością (czyli strumieniowych). W obu rozwiązaniach (z algorytmem PFQ i PDRR) konieczne jest utrzymywanie listy przepływów aktywnych AFL (*Active Flow List*). W przypadku algorytmu PDRR struktura ta utrzymuje informacje o identyfikatorach aktywnych przepływów, aktualnej wielkości licznika DC_i, kwantu przepływu Q_i oraz znacznikach umożliwiających realizację wyboru pakietu do transmisji. Selekcja pakietu do wysłania odbywa się przez cykliczne przeglądanie listy AFL. Pierwszeństwo ma jednak zawsze kolejka priorytetowa. Wielkość parametru *ByteCount(i)* decyduje o umieszczeniu pakietu w kolejce priorytetowej lub zwykłej. Jeśli *ByteCount(i)* jest mniejszy niż Q_i pakiety kierowane są do kolejki priorytetowej.

Jednym z podstawowych założeń FAN jest utrzymanie interfejsów użytkowników na dotychczasowym poziomie skomplikowania przy jednoczesnym umożliwieniu korzystania z usług o gwarantowanej jakości. Osiągnięcie wyżej założonego celu jest możliwe dzięki zastosowaniu mechanizmów sterowania dostępem do łącza oraz selekcji we wszystkich węzłach sieci. Zarządzanie listą chronionych przepływów przez poszczególne węzły pozwala na ograniczenie roli użytkowników w podejmowaniu decyzji dotyczących transmisji danych przepływów.

Jedną z głównych zalet sieci FAN jest podjęcie działań mających na celu utrzymanie jakości transmisji pakietów należących do obsługiwanych przepływów nawet w chwili wystąpienia awarii. FAN nie tylko zapewnia te same mechanizmy odpornościowe, które są stosowane we współczesnym Internecie, ale również możliwości korzystania z dodatkowych rozwiązań. Mechanizm sterowania dostępem do łącza zapewnia efektywne wykorzystanie ograniczonego pasma w przypadku wystąpienia awarii. Jest oczywiste, że nawet w dobrze zaplanowanej sieci, w chwili awarii przepustowość zostaje ograniczona. Może dojść do sytuacji, że ciągle napływający ruch nie będzie mógł być w całości obsłużony. Blok sterowania dostępem do łącza zapewnia, że przepływy, których transmisja już się rozpoczęła będą obsłużone bez utraty jakości. W przypadku ograniczonej przepustowości, mechanizm ten odrzuca bowiem pakiety nowych przepływów nie zarejestrowanych na liście. Możliwe jest również zarezerwowanie pewnej przepustowości dla usług specjalnych, jak na przykład do wykorzystania w czasie kataklizmów, czy w sytuacji, gdy klient płaci specjalnie za trwałą dostępność łącza itp. Istotne jest opracowanie mechanizmów ułatwiających dostęp do łącza dla przepływów strumieniowych. Zagadnienie to, stanowiące jeden z najważniejszych problemów do rozwiązania w sieciach FAN, jest przedmiotem prezentowanej rozprawy. Mechanizmy okresowego czyszczenia listy PFL oraz krótkotrwałego blokowania przepływów elastycznych aktywnych przez długi okres czasu mają za zadanie zapewnienie szybkiego dostępu nowego przepływu strumieniowego do łącza będącego w stanie natłoku. Rozwiązanie to jest istotne np. w przypadku rozmowy telefonicznej. Użytkownik próbujący nawiązać połączenie przyzwyczajony jest do krótkiego czasu oczekiwania na rozpoczęcie rozmowy w tradycyjnej telefonii (rzędu kilku sekund). Podobne wymagania należy spełnić projektując usługę telefonii w sieciach teleinformatycznych. W oryginalnym rozwiązaniu sieci FAN, w sytuacji natłoku, żądania nawiązania połączenia głosowego mogą być odrzucane, aż do momentu ustąpienia natłoku, a więc przez długi okres czasu, nieakceptowany przez użytkownika.

Oczywiście zaproponowane do zbadania mechanizmy nie mogą w znaczący sposób wpływać na transmisję pozostałego ruchu sieciowego. Zaletą wprowadzenia sieci FAN jest możliwość transmisji odrzuconych przepływów inną trasą. Identyfikatory ruchu, ściśle związane z trasą, powinny być dobierane za pomocą specjalnej funkcji indeksującej (*hash function*). W chwili odrzucenia pakietu, rozwiązanie takie pozwala na skorzystanie z alternatywnej trasy przez zmianę etykiety

przepływu. Jednym z podstawowych założeń FAN jest kompatybilność z usługami korzystającymi z transmisji niegwarantowanej, szeroko stosowanymi w Internecie. Istnieje także możliwość rozwoju nowych aplikacji, które mogą być ukierunkowane na pracę w czasie rzeczywistym. Ponadto pakiety poszczególnych aplikacji mogą być przesyłane przy użyciu wielu przepływów jednocześnie, co daje możliwość uzyskania większych przepływności.

Mechanizmy stosowane w FAN pozwalają na swobodę w doborze protokołu transportowego dla usług typu elastycznego. Daje to np. możliwość korzystania jednocześnie z wielu wersji protokołu TCP. Kolejnymi korzyściami płynącymi z zastosowania *Flow-Aware Networking* są większa efektywność oraz łatwiejsze rozliczanie klientów za korzystanie z sieci. Lepsze wykorzystanie dostępnych przepustowości dzięki mechanizmowi sterowania dostępem do łącza (szczególnie w sytuacji wystąpienia natłoku w sieci) pozwala na zmniejszenie kosztów użytkowania sieci. Z kolei brak klasyfikowania usług powoduje, że pakiety są widziane jako jednakowe z punktu widzenia ich zliczania. W FAN nie ma konieczności zawierania skomplikowanych kontraktów ruchowych. Nie występuje więc potrzeba przesyłania pakietów sygnalizacyjnych, a wszystkie obsługiwane pakiety mają za zadanie zapewnienie wykonania określonej użytecznej usługi. Z funkcjonalności FAN można korzystać również w sieciach używających innych architektur QoS. Nadaje się ona np. do sterowania przepływami, które w nadrzędnej architekturze korzystają z transmisji niegwarantowanej. Nie koliduje to z gwarancjami QoS jakie przyrzeczono użytkownikom korzystającym np. z mechanizmu *IntServ*, polepsza za to znacznie jakość obsługi przepływów strumieniowych nie korzystających z żadnych gwarancji. Koncepcja FAN gwarantowania jakości obsługi jest bardzo obiecująca. Mechanizmy stosowane w FAN nie są skomplikowane, a pozwalają na lepsze sterowanie podziałem dostępnej przepływności. Oczywistym jest więc również zmniejszenie kosztów utrzymania takiej sieci. Brak konieczności określania przez węzły a priori parametrów nadawanych przepływów daje możliwość wprowadzenia FAN praktycznie bez zmian w aplikacjach. Rezygnacja z klas usług stosowanych z reguły dla zapewnienia jakości jest niepodważalną zaletą FAN, która nie tylko upraszcza samą strukturę sieci, ale umożliwia np. łatwiejszy sposób naliczania opłat za korzystanie z sieci. Jak w każdym nowym rozwiązaniu, tak i w przypadku FAN, wiele aspektów, głównie technicznych, wymaga jeszcze dopracowania. W szczególności chodzi o realizację samych ruterów, modyfikacje usług na potrzeby FAN, itp. Bardzo dobra skalowalność i natychmiastowa skuteczność (ze względu na samowystarczalność – węzeł FAN sam steruje się przy pomocy parametrów, które wyznacza) sprawiają, że *Flow-Aware Networking* wychodzi naprzeciw wymaganiom współczesnego Internetu. Jest rozwiązaniem mającym wiele zalet, a przede wszystkim możliwym do wprowadzenia w stosunkowo niedługim czasie.

Ważną zaletą sieci FAN jest wspomniana skalowalność [6,7]. Dzięki zastosowaniu bloku sterowania dostępem oraz buforów o odpowiednio dobranych rozmiarach możliwe jest stosowanie ruterów wzajemnie zabezpieczonych nawet w sieciach o bardzo szybkich łączach. Istotne jest zauważenie faktu, że liczba aktywnych przepływów (a więc takich, które posiadają pakiety w kolejce) w dowolnej chwili czasu jest ograniczona.

Kolejną zaletą sieci FAN jest ich neutralność. Zagadnienie sieci neutralnych (*net neutrality*) jest bardzo istotne w ostatnich latach. Odpowiednie prawo, nad którym prowadzone są obecnie prace, m.in. w Kongresie Stanów Zjednoczonych Ameryki, ma zapewnić równe traktowanie użytkowników bez względu np. na stosowane aplikacje. Obecnie operator może różnicować ruch w sposób jawny, zapewniając przykładowo lepsze warunki obsługi ruchu wysłanego przez aplikację inną niż zalecana przez dostawcę łącza. Działanie takie jest niekorzystne dla użytkownika oraz producentów oprogramowania. W konsekwencji takich działań utrudniona może stać się uczciwa konkurencja, może powstać Internet dwóch jakości oraz mogą być blokowane propozycje małych firm. FAN wychodzi naprzeciw tym problemom. Niejawna klasyfikacja ruchu odbiera dostawcom łączy możliwości ingerencji w jego obsługę na poziomie aplikacji, a tym samym sprawia, że sieci FAN mają cechy wskazane pojęciem *net neutrality*.

Zakres tematyczny rozprawy

W pierwszej części rozprawy zaproponowano cztery mechanizmy sterowania przeciążeniami w sieci FAN [1,2]. Są to:

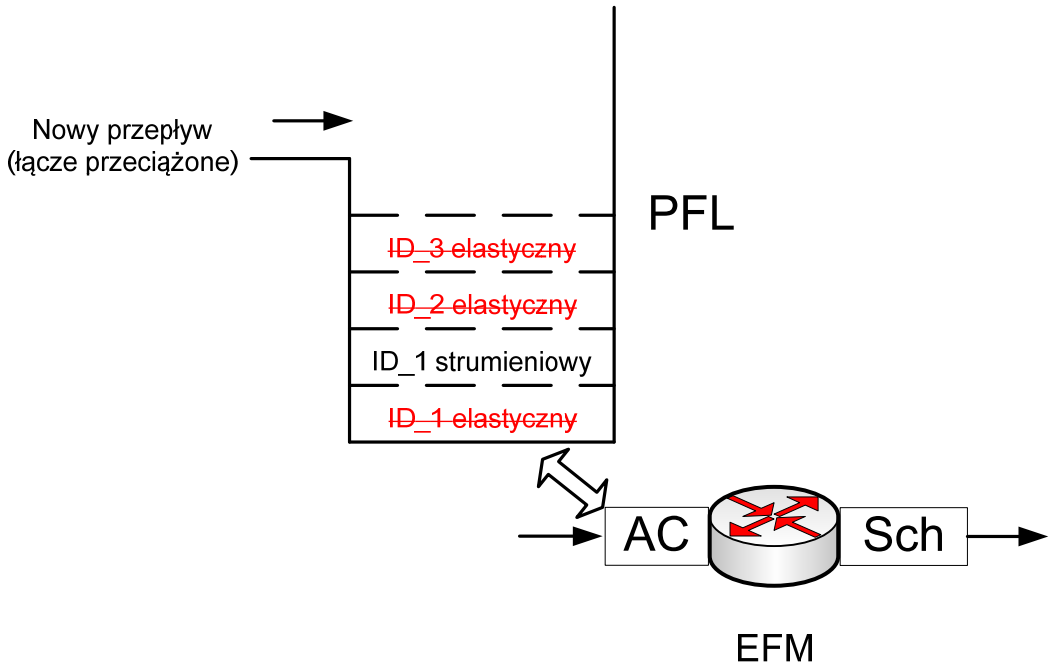
- EFM (*Enhanced Flushing Mechanism*),
- RAEF (*Remove Active Elastic Flows*),
- RBAEF (*Remove and Block Active Elastic Flows*),
- RPAEF (*Remove and Prioritize in access Active Elastic Flows*).

Zasada działania wszystkich mechanizmów jest związana z okresowym całkowitym bądź częściowym czyszczeniem listy przepływów chronionych (PFL) w bloku sterowania dostępem. Podstawowym celem ich działania jest zmniejszenie czasów oczekiwania na akceptację w bloku sterowania dostępem przez przepływy strumieniowe (np. połączenia VoIP). Okazuje się, że akceptowalny z punktu widzenia użytkownika czas oczekiwania na połączenie lokalne powinien być mniejszy od 6 s. Dla połączeń międzynarodowych analogiczny czas wynosi 11 s [5]. W podstawowej wersji sieci FAN, nowe przepływy nie mogą być zaakceptowane w sytuacji wystąpienia natłoku. Możliwe jest więc, że połączenia głosowe będą musiały czekać na akceptację przez długi, nieakceptowany okres czasu. Zaproponowane mechanizmy nie tylko zmniejszają czas oczekiwania na

akceptację przez przepływy strumieniowe, ale także pozwalają to uczynić bez zwiększania czasu transmisji pozostałego ruchu w sieci.

Mechanizm EFM

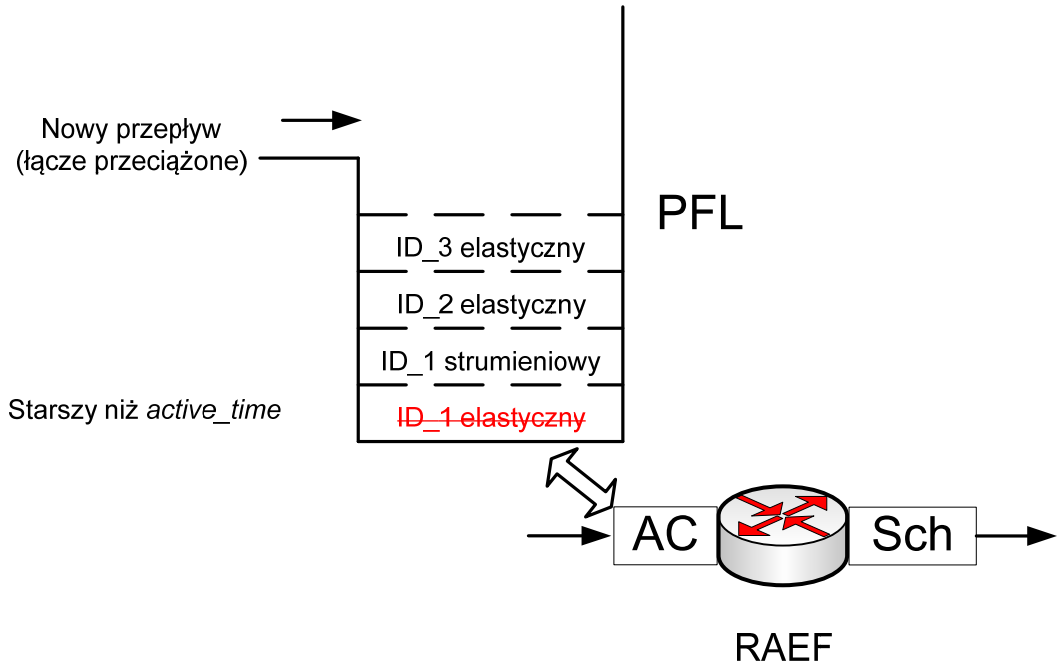
Mechanizm EFM (*Enhanced Flushing Mechanism*) został zaproponowany w celu rozwiązania problemu zbyt długich czasów oczekiwania na akceptację w ruterze przez przepływy strumieniowe. W rozwiązaniu tym, gdy pakiet nowego przepływu dociera do rutera w sytuacji wystąpienia przeciążenia, z listy PFL usuwane są identyfikatory wszystkich przepływów elastycznych. Proces ten jest zobrazowany na Rys. 2. Podstawowym parametrem, stosowanym w tym rozwiązaniu jest *pfl_flushing_timer*. Wartość tego wskaźnika oznacza minimalny przedział czasu, jaki musi upłynąć między dwoma następującymi po sobie akcjami czyszczenia listy PFL. Wprowadzenie tego parametru jest spowodowane koniecznością zapewnienia stabilnej pracy algorytmu EFM. Przepływy, których identyfikatory zostają usunięte w trakcie czyszczenia listy PFL nie są blokowane i mogą zostać zaakceptowane ponownie w ruterze w krótkim czasie. Jednakże usunięte przepływy muszą ponownie rywalizować o dostęp do łącza z kolejnymi przepływami, które również chcą rozpocząć transmisję. Mechanizm EFM może pracować w obu znanych rozwiązaniach rutera wzajemnie zabezpieczonego (z algorytmem PFQ lub PDRR). Implementacja tego rozwiązania w ruterze jest prosta i nie zwiększa w sposób istotny jego złożoności ani nie wymaga znaczącego zwiększenia wymaganych zasobów.



Rys. 2. Schemat działania mechanizmu EFM

Mechanizm RAEF

W mechanizmie RAEF (*Remove Active Elastic Flows*) w chwili przybycia do rutera pakietu nowego przepływu, z listy PFL usuwane są identyfikatory przepływów elastycznych, które są aktywne przez określony czas zadany parametrem *active_time*. Schemat działania tego algorytmu jest przedstawiony na Rys. 3. Podobnie jak w mechanizmie EFM, usunięte przepływy nie są blokowane i mogą natychmiast po czyszczeniu listy rywalizować z innymi przepływami o ponowny dostęp do łącza. Podobnie także jak w poprzednim przypadku, algorytm RAEF działa dobrze w dowolnej znanej architekturze rutera wzajemnie zabezpieczonego. Obciążenie rutera związane z ilością wykonywanych obliczeń, a także zużyte zasoby pozostają również na zadowalającym poziomie.

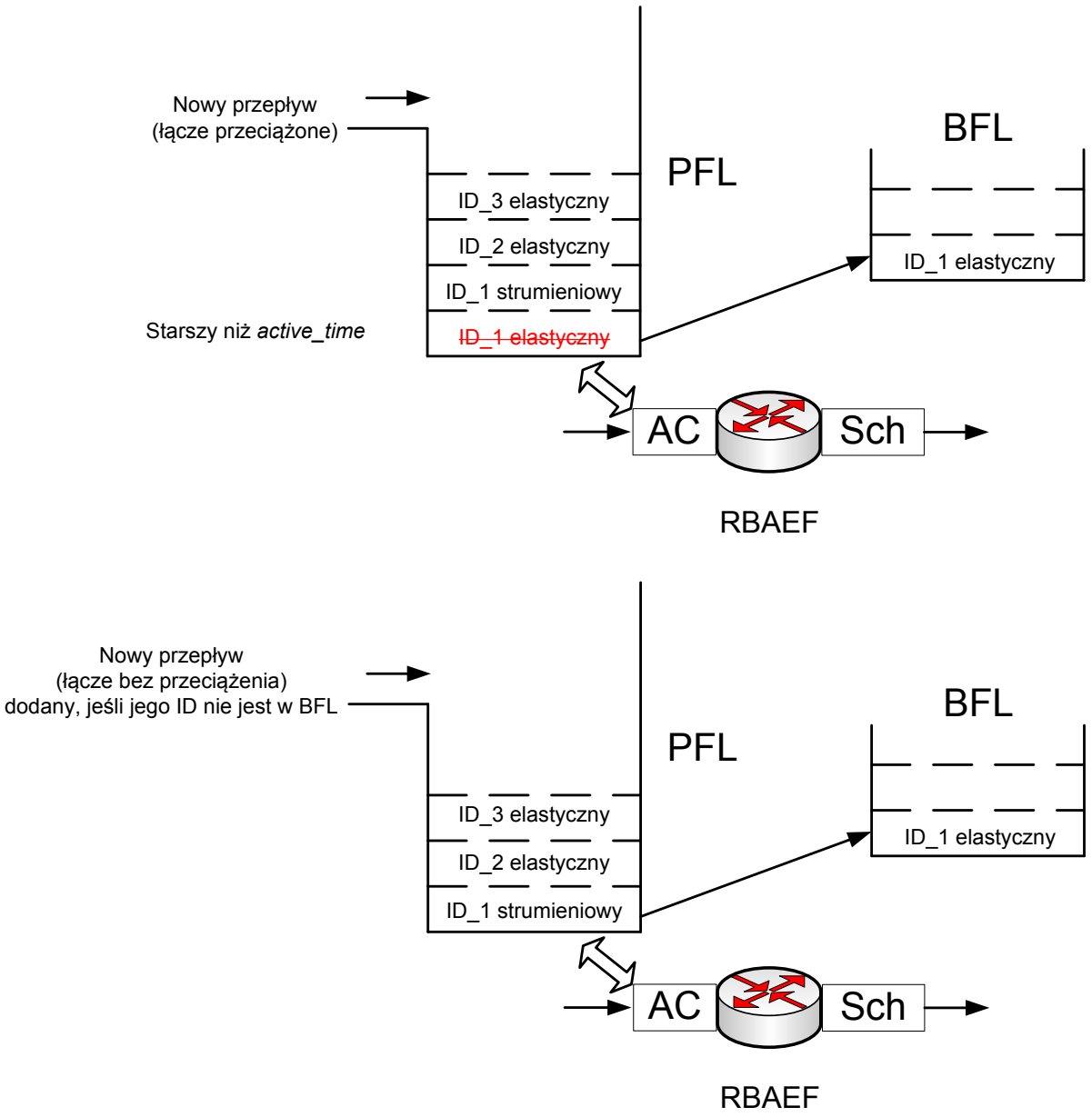


Rys. 3. Schemat działania mechanizmu RAEF

Mechanizm RBAEF

Mechanizm RBAEF (*Remove and Block Active Elastic Flows*) jest rozszerzoną wersją mechanizmu RAEF. Jego schemat działania jest przedstawiony na Rys. 4. W rozwiązaniu tym, w chwili nadejścia pakietu nowego przepływu z listy PFL usuwane są identyfikatory przepływów elastycznych aktywnych przez zadaną parametrem *active_time* ilość czasu. Różnica w stosunku do algorytmu RAEF polega na tym, że identyfikatory usuniętych przepływów są zapisywane na liście przepływów blokowanych BFL (*Blocked Flow List*) na krótki okres czasu zadany parametrem *blocked_time*. Nowy przepływ nie może być zaakceptowany w bloku sterowania dostępem, jeśli jego identyfikator jest

zapisany na liście BFL. Podejście takie pozwala na zmniejszenie liczby przepływów rywalizujących o dostęp do łącza po akcji czyszczenia listy PFL, a tym samym na zmniejszenie liczby nowo zaakceptowanych przepływów. Podobnie jak oba prezentowane wcześniej rozwiązania, algorytm RBAEF działa w zadowalający sposób we wszystkich znanych architekturach rutera wzajemnie zabezpieczonego. Złożoność obliczeniowa jak i wykorzystanie zasobów są w tym przypadku większe niż we wcześniejszych propozycjach, jednak w pełni akceptowalne z punktu widzenia obciążenia rutera.



Rys. 4. Schemat działania mechanizmu RBAEF

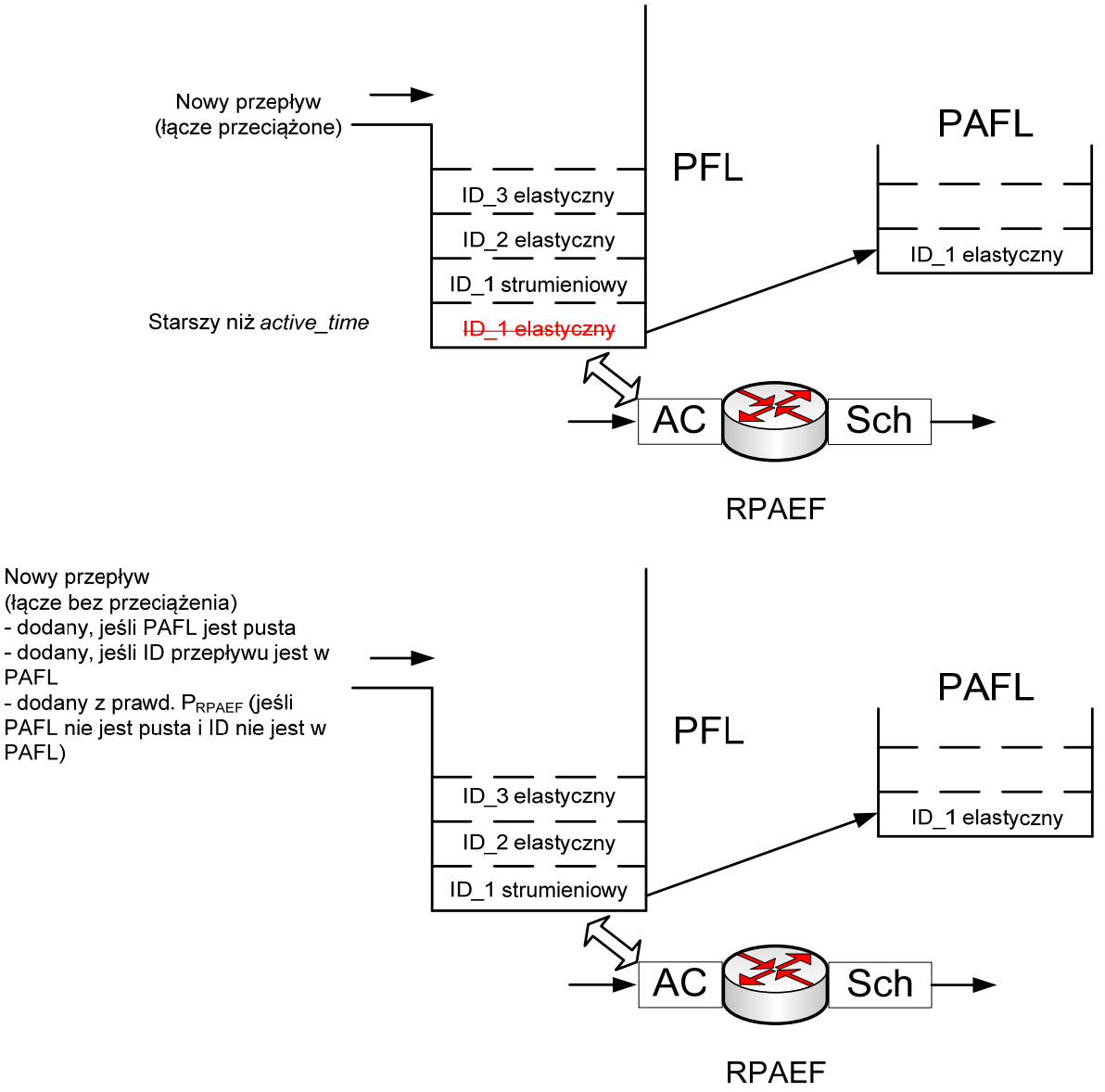
Mechanizm RPAEF

Ostatni zaproponowany w rozprawie mechanizm sterowania przeciążeniami w sieci FAN to RPAEF (*Remove and Prioritize in access Active Elastic Flows*). Celem tego rozwiązania jest wyeliminowanie niedogodności towarzyszących implementacji mechanizmu RBAEF. Chodzi tu przede wszystkim o czas transmisji przepływów elastycznych. Nawet krótki czas blokowania ich po akcji czyszczenia listy PFL może spowodować znaczące wydłużenie czasu transmisji. W mechanizmie RPAEF listę BFL zastąpiono listą PAFL (*Priority Access Flow List*), a więc listą przepływów posiadających wysoki priorytet w czasie rywalizowania o dostęp do łącza. Schemat działania mechanizmu RPAEF jest przedstawiony na Rys. 5. W sytuacji braku przeciążenia, nowe przepływy są zawsze akceptowane. W momencie wystąpienia natłoku, nadejście pakietu nowego przepływu powoduje usunięcie z listy PFL identyfikatorów przepływów aktywnych przez czas równy co najmniej wartości parametru *active_time*. Usunięte identyfikatory są zapisywane na liście PAFL i pozostają tam przez krótki ustalony okres czasu. Gdy lista PAFL nie jest pusta, nowe przepływy są akceptowane z małym prawdopodobieństwem P_{RPAEF}. Z drugiej strony, przepływy, których identyfikatory znajdują się na liście PAFL są akceptowane z prawdopodobieństwem równym 1. Tym sposobem identyfikatory odrzuconych przepływów są dodawane ponownie do listy PAFL w krótkim czasie. Jednocześnie nowe przepływy mają szansę na szybsze skorzystanie z zasobów. Stopień skomplikowania implementacji mechanizmu RPAEF jest taki sam jak w przypadku mechanizmu RBAEF.

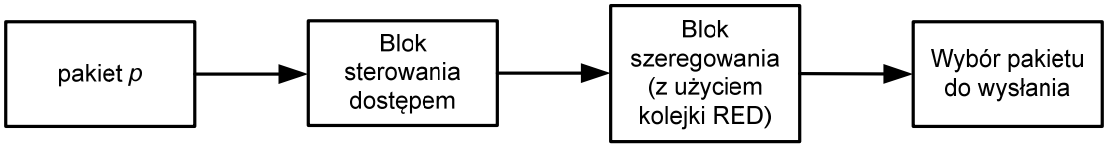
W drugiej części rozprawy zaproponowano nową architekturę sieci FAN, nazwaną AFAN (*Approximate FAN*). Podstawowym elementem nowego rozwiązania jest ruter wzajemnie zabezpieczony z użyciem algorytmu RED (*Random Early Detection*) w celu zapewnienia sprawiedliwej obsługi ruchu elastycznego w ruterze oraz wysokiego priorytetu dla pakietów należących do przepływów strumieniowych.

W rozwiązaniu tym stosuje się jedynie dwie kolejki FIFO, po jednej dla pakietów przepływów elastycznych i przepływów strumieniowych. Nie ma konieczności utrzymywania listy AFL, a więc listy przepływów aktywnych, ani żadnych dodatkowych elementów względem dwóch znanych rozwiązań, z algorytmem PFQ lub PDRR. Nowe rozwiązanie zapewnia funkcjonalność sieci FAN, zgodnie z jej założeniami przy jednoczesnym zmniejszeniu obciążenia rutera związanego z ilością dokonywanych obliczeń.

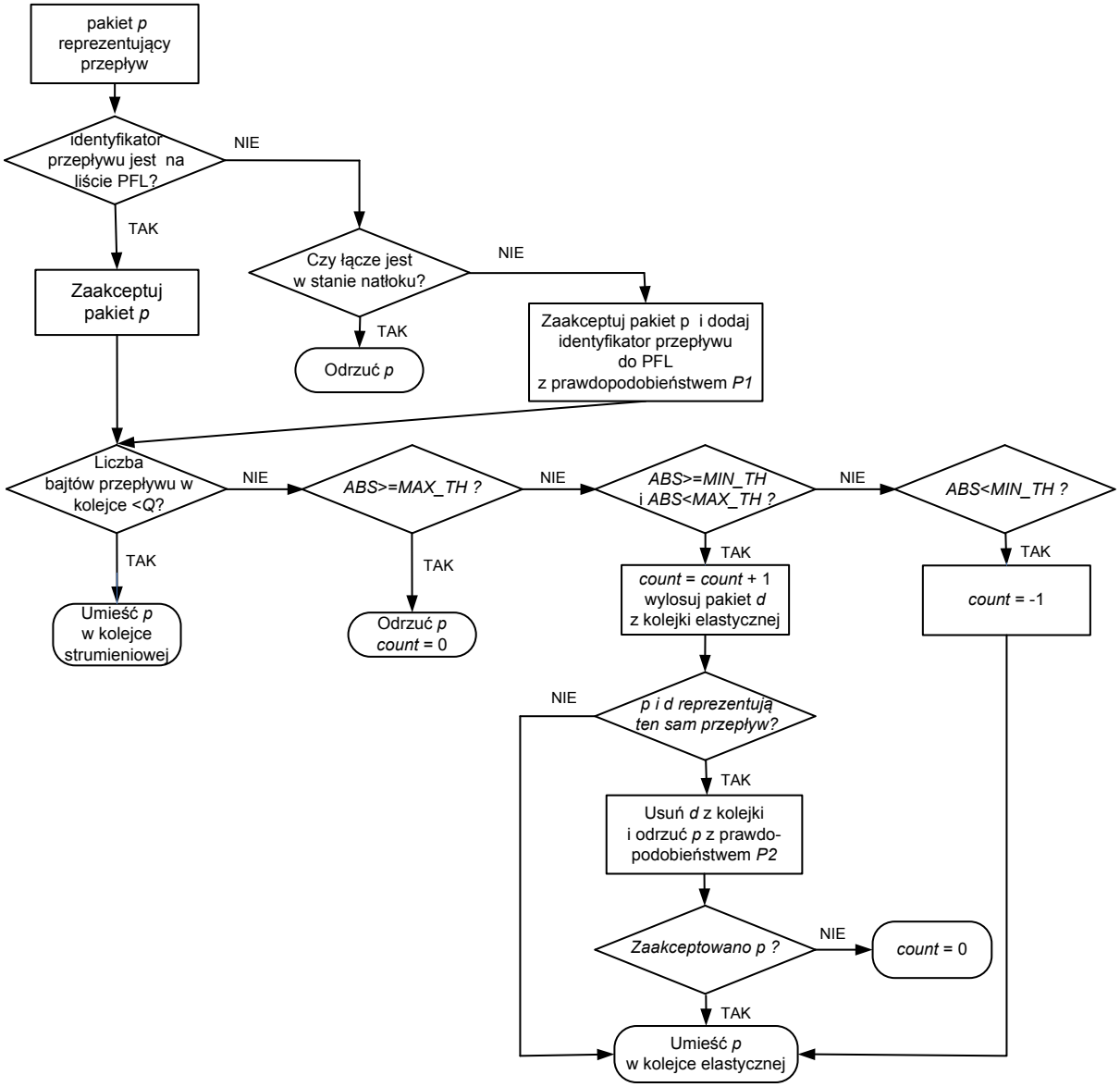
Obsługa pakietu w nowej propozycji jest przedstawiona przy użyciu schematu blokowego ilustrującego działanie rutera (Rys. 6), algorytmu działania (Rys. 7) oraz pseudokodu umożliwiającego implementację algorytmu w ruterze, zaprezentowanego w Tab. 1.



Rys. 5. Schemat działania mechanizmu RPAEF



Rysunek 6.
Schemat blokowy obsługi pakietów w ruterze FAN z algorytmem RED



Rysunek 7.
Algorytm realizacji kolejkowania pakietów w ruterze FAN z kolejką RED

Dowolny pakiet *p* przybywający do rutera wzajemnie zabezpieczonego z algorytmem RED jest analizowany w bloku sterowania dostępem. Jeżeli identyfikator przepływu reprezentowanego przez pakiet *p* jest zapisany na liście PFL, to pakiet jest akceptowany i przesłany do bloku szeregowania rutera (wiersze 2-3 w Tab. 1). Jeśli pakiet *p* reprezentuje nowy przepływ (dotychczas niezaakceptowany) to może on być zaakceptowany jedynie w przypadku braku natłoku w łączu. Wówczas pakiet jest przesyłany do bloku szeregowania, a identyfikator przepływu jest zapisywany w PFL z prawdopodobieństwem *P1* (wiersz 6 w Tab. 1).

Tabela 1.

Pseudokod umożliwiający realizację rutera FAN z kolejką RED

Moduł bloku sterowania dostępem:
1. dla pakietu *p* przychodzącego do rutera
2. **If** *flow_ID* _ PFL **then**
3. zaakceptuj pakiet *p*
4. **Else If** łącze jest przeciążone **then**
5. odrzuć pakiet *p*
6. **Else** zaakceptuj pakiet *p* z prawdopodobieństwem *P1*

Moduł kolejkowania:
7. dla pakietu *p* zaakceptowanego w bloku sterowania dostępem
8. **If** flow bytes <= Q **then**
9. *Enqueue*(*PQ*; *p*) – umieść pakiet *p* w kolejce priorytetowej
10. *flow_bytes* = *flow_bytes* + *size*(*p*)
11. **Else** oblicz *ABS* dla kolejki FIFO przepływów elastycznych
12. **If** *ABS* >= *MAX_TH* **then**
13. odrzuć pakiet *p*
14. *count* = 0
15. **If** *MIN_TH* <= **ABS** < *MAX_TH* **then**
16. *count* = *count* + 1
17. wylosuj pakiet *d* z kolejki FIFO dla przepływów elastycznych
18. **If** *flow_ID* powiązane z pakietem *p* = *flow_ID* powiązane z pakietem *d* **then**
19. usuń pakiet *d* z kolejki
20. oblicz prawdopodobieństwo *P2*
21. odrzuć pakiet *p* z prawdopodobieństwem *P2*
22. **If** *p* zostaje odrzucony **then**
23. *count* = 0
24. **Else** *p* zostaje przekazany do umieszczenia w kolejce
25. **Else** *p* zostaje przekazany do umieszczenia w kolejce
26. **If** *ABS* < *MIN_TH* **then**
27. *p* zostaje przekazany do umieszczenia w kolejce
28. *count* = -1
29. **If** *flow_bytes* > *Q* oraz *p* zostaje przekazany do umieszczenia w kolejce **then**
30. *Enqueue*(*EQ*; *p*) – umieść pakiet *p* w kolejce elastycznej
31. *flow_bytes* = *flow_bytes* + *size*(*p*)

Moduł selekcji pakietu do wysłania:
32. **While** (*PQ* jest niepusta) **do**
33. *p* = *Dequeue*(*PQ*) – wybierz pakiet *p* z kolejki priorytetowej
34. *Send*(*p*)
35. *flow_bytes* = *flow_bytes* - *size*(*p*)
36. **If** kolejka FIFO elastyczna nie jest pusta **then**
37. *p* = *Dequeue*(*EQ*) – wybierz pakiet *p* z kolejki elastycznej
38. *Send*(*p*)
39. *flow_bytes* = *flow_bytes* - *size*(*p*)

Zaletą tego rozwiązania jest obliczanie wartości parametrów *fair_rate* oraz *priority_load* w sposób mniej skomplikowany niż w przypadku dwóch znanych rozwiązań (z algorytmem PFQ lub PDRR). Wartości parametru *fair_rate* są obliczane ze wzoru (3), jednakże *FB* reprezentuje w tym przypadku liczbę bajtów wysłanych przez przepływy elastyczne w czasie $t_2 - t_1$ podzieloną przez liczbę przepływów, których identyfikatory są zapisane w PFL. Podejście takie daje przybliżony wynik *fair_rate*, ale jest wystarczająco dokładne, by stosować je do estymacji wartości tego parametru. Wartości parametru *priority_load* są wyznaczane ze wzoru (2). W tym przypadku zliczana jest jednak liczba bajtów wysłanych przez przepływy strumieniowe w obserwowanym przedziale czasu, co lepiej oddaje rzeczywisty charakter tego parametru.

Operacja kolejkowania przychodzącego pakietu stanowi najważniejszy element nowego rozwiązania. Jeśli przychodzący pakiet reprezentuje przepływ mający w kolejce mniej niż ustalona wartość kwantu *Q* (zwykle ustawiane na wartość MTU) bajtów, to jest on umieszczany w kolejce priorytetowej (wiersze 7-10 w Tab. 1). W przeciwnym przypadku konieczne jest wyznaczenie wartości parametru przybliżonej zajętości bufora *ABS* (*Approximate Buffer Size*) z poniższego wzoru:

$$\begin{cases} ABS = (1 - w_q)ABS + w_q q & \text{jeśli kolejka jest niepusta} \\ ABS = (1 - w_q)^m ABS & \text{jeśli kolejka jest pusta} \end{cases} \quad (4)$$

gdzie w_q jest wagą kolejki, *q* reprezentuje aktualną wielkość bufora, a *m* jest liczbą pakietów, które mogłyby być wysłane przez ruter w czasie, gdy łącze jest wolne. Wartości parametru *m* są wyznaczane z poniższego wzoru:

$$m = (time - q\_time) / s \quad (5)$$

gdzie *time* jest aktualnym czasem, *q_time* jest czasem początku okresu bezczynności, a *s* jest czasem transmisji pojedynczego pakietu. Jeśli *ABS* jest większe lub równe maksymalnej dopuszczalnej wartości progu dla bufora (*MAX_TH*), przychodzący pakiet musi być usunięty, a licznik *counter* (mierzący liczbę przychodzących pakietów od ostatniej operacji usunięcia pakietu) jest ustawiany na wartość 0 (wiersze 12-14 w Tab. 1). Jeśli wartość parametru *ABS* jest mniejsza niż *MAX_TH*, ale większa lub równa od wartości drugiego progu ustawionego dla bufora (*MIN_TH*), licznik *count* jest zwiększany o 1. W tej sytuacji z kolejki FIFO dla przepływów elastycznych losowany jest pakiet *d*. Identyfikatory przepływów reprezentowanych przez pakiety *p* i *d* są ze sobą porównywane (wiersze 15-17 w Tab. 1). Losowanie pakietu z kolejki polega na wygenerowaniu liczby losowej i

przyporządkowaniu jej bajtowi w kolejce. Następnie wylosowany bajt jest przyporządkowywany pakietowi. Działanie takie pozwala na poprawne działanie rutera w sytuacji transmisji pakietów różnej długości. Jeśli oba pakiety (*p* i *d*) reprezentują ten sam przepływ, pakiet *d* jest usuwany z kolejki, a pakiet *p* jest odrzucany z prawdopodobieństwem *P2* (wiersze 18-21 w Tab. 1). Jeśli pakiet *p* jest odrzucony, licznik c*ount* jest zerowany. W przeciwnym przypadku pakiet *p* zostaje umieszczony w kolejce (wiersze 22-24 w Tab. 1). Pakiet *p* może być także umieszczony w kolejce w sytuacji, gdy porównywane identyfikatory są różne lub *ABS* jest mniejszy niż *MIN_TH* (wiersze 25-27 w Tab. 1). W ostatnim przypadku *count* jest ustawiany na wartość −1, czyli wartość początkową (wiersz 28 w Tab. 1). Jeśli liczba bajtów przepływu reprezentowanego przez pakiet *p* w buforze jest większa niż *Q*, pakiet może być umieszczony jedynie w kolejce FIFO dla przepływów elastycznych (wiersze 29-31 w Tab. 1). Sposób umieszczania pakietu w kolejce elastycznej jest zgodny z rozwiązaniem AFD (*Approximate Fair Dropping*), czyli algorytmem sprawiedliwego umieszczania pakietów w kolejce.

Proces wyboru pakietu do wysłania jest bardzo prosty. Jeśli kolejka priorytetowa jest niepusta, wybierany jest pierwszy pakiet z kolejki (wiersze 32-35 w Tab. 1). W przeciwnym przypadku wybierany jest pierwszy pakiet z kolejki przeznaczonej na pakiety przepływów elastycznych (wiersze 36-39 w Tab. 1).

W proponowanym ruterze istotne jest dobranie właściwych wielkości poszczególnych parametrów. Wartości parametru w_q powinny zawierać się w przedziale 0.001-0.0042. Wielkość ta reprezentuje stałą czasową filtru dolnoprzepustowego (rozwiązanie zaczerpnięte z algorytmu RED). Wartości *MIN_TH* oraz *MAX_TH* są kluczowe dla prezentowanego rozwiązania. Decydują one o obciążeniu bufora oraz zapewniają sprawiedliwą i wydajną transmisję w łączu. Różnica *MAX_TH-MIN_TH* powinna być większa niż średni wzrost wartości *ABS* w czasie równym szybkości przesłania pakietu w łączu do miejsca docelowego i z powrotem (*roundtrip time*). W praktyce *MAX_TH* powinien być przynajmniej dwukrotnie większy niż *MIN_TH*. Wartość prawdopodobieństwa *P2* jest obliczana z poniższego wzoru:

$$P2 = P2_{temp} /(1 - count \cdot P2) \tag{6}$$

$P2_{temp}$ jest obliczane w funkcji parametru *ABS* i wyznaczane z poniższego wzoru:

$$P2_{temp} = MAX_p (ABS - MIN\_TH) /(MAX\_TH - MIN\_TH) \tag{7}$$

Gdzie MAX_p jest maksymalną dopuszczalną wartością parametru $P2_{temp}$. Wartość MAX_p powinna być ustawiona w sposób umożliwiający wolne zmiany prawdopodobieństwa *P2* (np. MAX_p =0.02).

Cel i teza rozprawy

Celem badań przedstawionych w niniejszej rozprawie jest kompleksowa analiza zaproponowanych mechanizmów sterowania przeciążeniami oraz nowej architektury AFAN. W tym celu przeprowadzone zostały badania symulacyjne przy użyciu symulatora sieciowego ns-2. Otrzymane wyniki wskazują na słuszność stosowania i potwierdzają zalety propozycji przedstawionych przez autora rozprawy.

Autor wysuwa następującą tezę:

Możliwe jest zdefiniowanie efektywnych i prostych mechanizmów sterowania przeciążeniami w sieciach zorientowanych na przepływy (Flow Aware Networks)

(ang. It is possible to define efficient and simple congestion control mechanisms in Flow Aware Networks)

Wykonane zadania

Praca ma charakter teoretyczno-symulacyjny. Część praktyczna obejmuje wykonanie szeregu symulacji komputerowych umożliwiających zbadanie wartości najbardziej istotnych parametrów transmisji w różnych scenariuszach. Wszystkie badania symulacyjne zostały przeprowadzone przy użyciu znanego i cenionego symulatora sieciowego ns-2 [10]. Proponowane rozwiązania zostały zaimplementowane w symulatorze. Uzyskane wyniki zostały opracowane przy użyciu języka programowania PERL oraz programu Excel. Wykresy zostały narysowane przy użyciu programu Gnuplot. Scenariusze symulacyjne zostały starannie wyselekcjonowane, a każde badanie było powtarzane dziesięć razy w tych samych warunkach. Umożliwiło to przeprowadzenie analizy statystycznej otrzymanych wyników.

Podstawowe parametry ustawiane w trakcie symulacji:

- liczba przepływów elastycznych: zmienna,
- odstępy czasu pomiędzy napływającymi przepływami: zgodne z rozkładem wykładniczym (średnia zmienna, najczęściej 0.1 s)
- wielkość danych do przesłania przez przepływy elastyczne: zgodna z rozkładem Pareto (parametry zmienne).

- przepływność badanego łącza: 100 Mbit/s
- okres pomiaru wartości *priority_load*: 50 ms
- okres pomiaru wartości *fair_rate*: 500 ms
- *max_priority_load*: 70%
- *min_fair_rate*: 5%
- czas „rozbiegu” symulacji: 20 s
- rozkład statystyczny do oceny błędu symulacji: t-Studenta z przedziałem ufności 95%

Streszczenie prezentuje najważniejsze osiągnięcia przedstawione w rozprawie. Pominięta została m.in. analiza związana z wprowadzeniem nowych metod obliczania parametrów *priority_load* oraz *fair_rate*, a także analiza przedstawiająca skuteczność zaproponowanych mechanizmów sterowania przeciążeniami w sytuacji wystąpienia awarii. Poniżej zaprezentowano najważniejsze wyniki przedstawiające zalety i wady mechanizmów EFM, RAEF, RBAEF i RPAEF w sieciach FAN z algorytmem PFQ oraz nowej architektury AFAN. Szczegółowa analiza działania mechanizmów sterowania przeciążeniami dla sieci FAN z algorytmem PDRR zawarta jest w [3]. W [4] można znaleźć wyniki badań opisywanych mechanizmów w sieciach FAN w sytuacji wystąpienia awarii.

Średni czas akceptacji przepływów strumieniowych w ruterze z zaimplementowanym mechanizmem EFM i algorytmem PFQ jest zaprezentowany na Rys. 8, natomiast średnia liczba zaakceptowanych przepływów w bloku sterowania dostępem jest przedstawiona na Rys. 9. Średni czas transmisji przepływów elastycznych w łączu pokazano na Rys. 10. Na wszystkich rysunkach zaznaczono 95% przedziały ufności.

Uzyskane wyniki wskazują, że akceptowalny czas przyjęcia nowych przepływów strumieniowych w łączu można osiągnąć dla *pfl_flushing_timer* = 5 s lub 10 s. Niestety w tym przypadku średnia liczba zaakceptowanych przepływów na liście PFL po każdej akcji jej czyszczenia jest zbyt wysoka w porównaniu z podstawową wersją sieci FAN. Podobnie, czas transmisji przepływów elastycznych ulega znacznemu wydłużeniu dla małych wartości parametru *pfl_flushing_timer*. Na Rys. 8 i 9 przedstawiono jedynie wartości dla algorytmu PFQ, gdyż dla PDRR były one podobne.

Krzywe z Rys. 10 można aproksymować poniższymi wzorami:

$y = -108.94 \ln x + 611.43$ dla algorytmu PFQ (8)

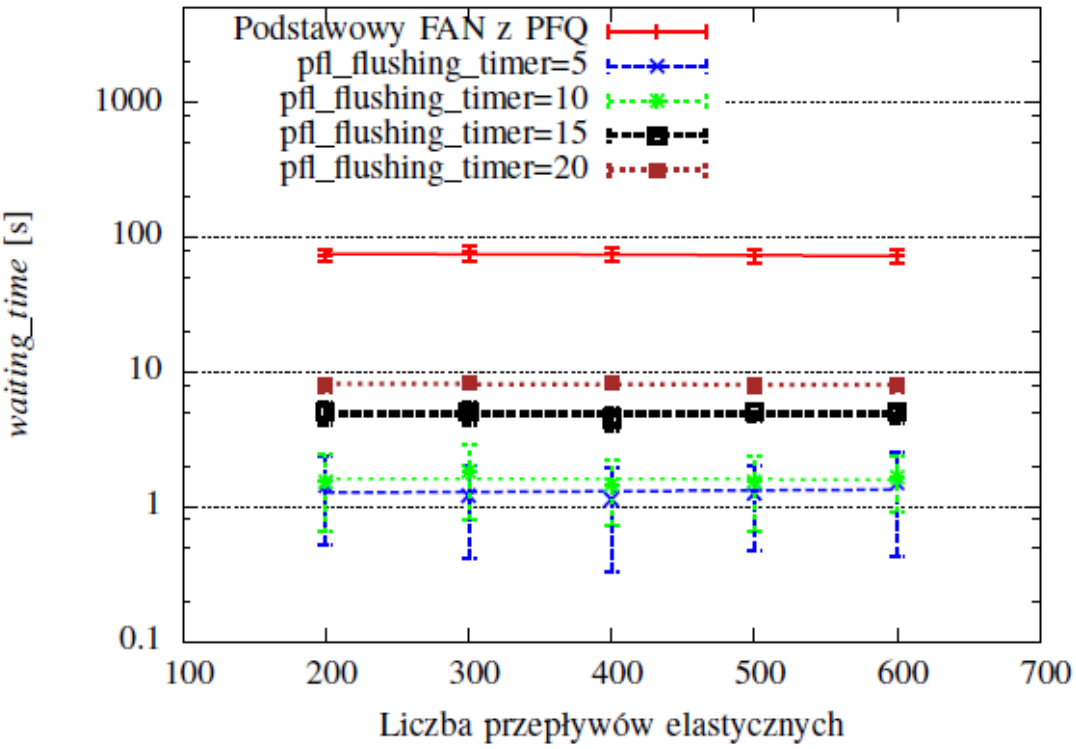
$y = -82.84 \ln x + 521.37$ dla algorytmu PDRR (9)

Korzystając ze wzorów (8) i (9) można wyznaczyć takie wartości parametru *pfl_flushing_timer*, dla których nie ulegnie wydłużeniu średni czas transmisji przepływów elastycznych. Są to odpowiednio wartości:

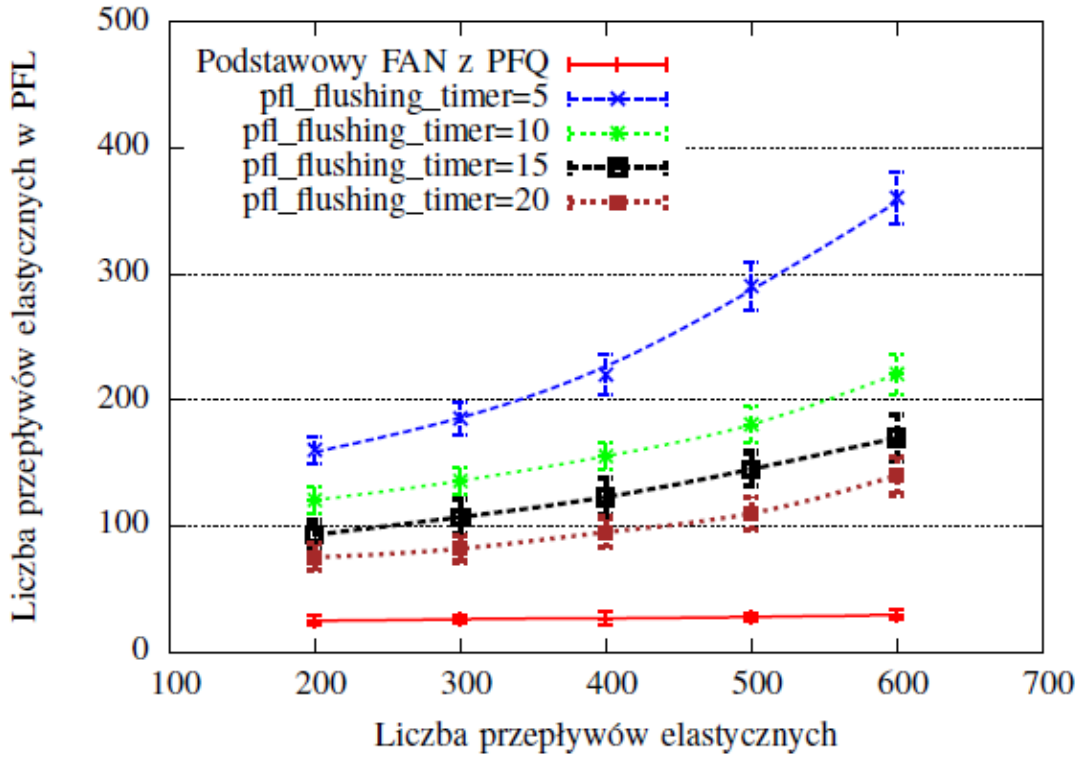
pfl_flushing_timer = 89.50 s (dla PFQ)

pfl_flushing_timer = 119.50 s (dla PDRR)

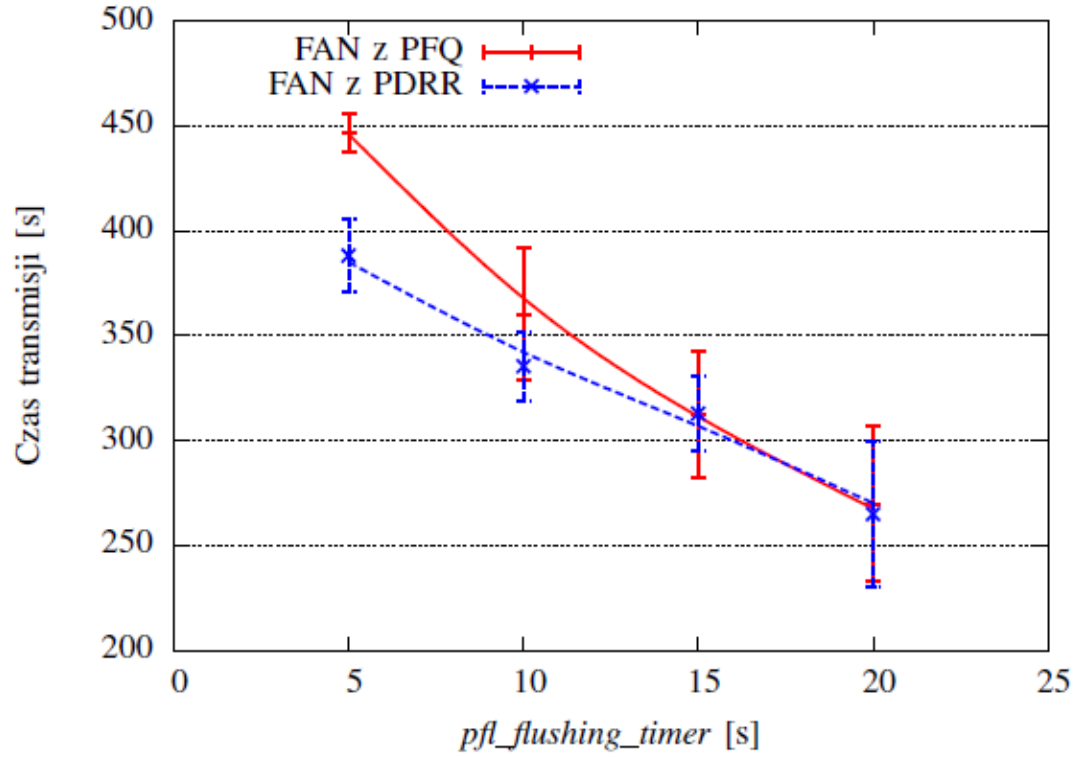
Niestety wartości te nie są akceptowalne z punktu widzenia czasu oczekiwania na dostęp do łącza przez przepływy strumieniowe.



Rys. 8. Średni czas akceptacji przepływów strumieniowych w łączu FAN z mechanizmem EFM



Rys. 9. Liczba przepływów elastycznych w PFL w łączu FAN z mechanizmem EFM



Rys. 10. Średni czas transmisji przepływów elastycznych w łączu FAN z mechanizmem EFM

Średni czas akceptacji przepływów strumieniowych w ruterze z zaimplementowanym mechanizmem RAEF i algorytmem PFQ jest zaprezentowany na Rys. 11, natomiast średnia liczba zaakceptowanych po akcji czyszczenia zawartości listy PFL przepływów w bloku sterowania dostępem jest przedstawiona na Rys. 12. Średni czas transmisji przepływów elastycznych w łączu pokazano na Rys. 13.

Uzyskane wyniki wskazują, że podobnie jak w przypadku mechanizmu EFM, odpowiedni czas akceptacji nowych przepływów strumieniowych w łączu można osiągnąć dla *active_time* = 5 s lub 10 s. Niestety, także i w tym przypadku średnia liczba zaakceptowanych przepływów na liście PFL jest zbyt wysoka w porównaniu z podstawową wersją sieci FAN. Także czas transmisji przepływów elastycznych ulega wydłużeniu dla małych wartości parametru *active_time*. Na Rys. 11 i 12 przedstawiono jedynie wartości dla algorytmu PFQ, gdyż dla PDRR były one podobne.

Krzywe z Rys. 13 można aproksymować poniższymi wzorami:

$$y = -10.25 \ln x + 464.69 \quad \text{dla algorytmu PFQ} \tag{10}$$

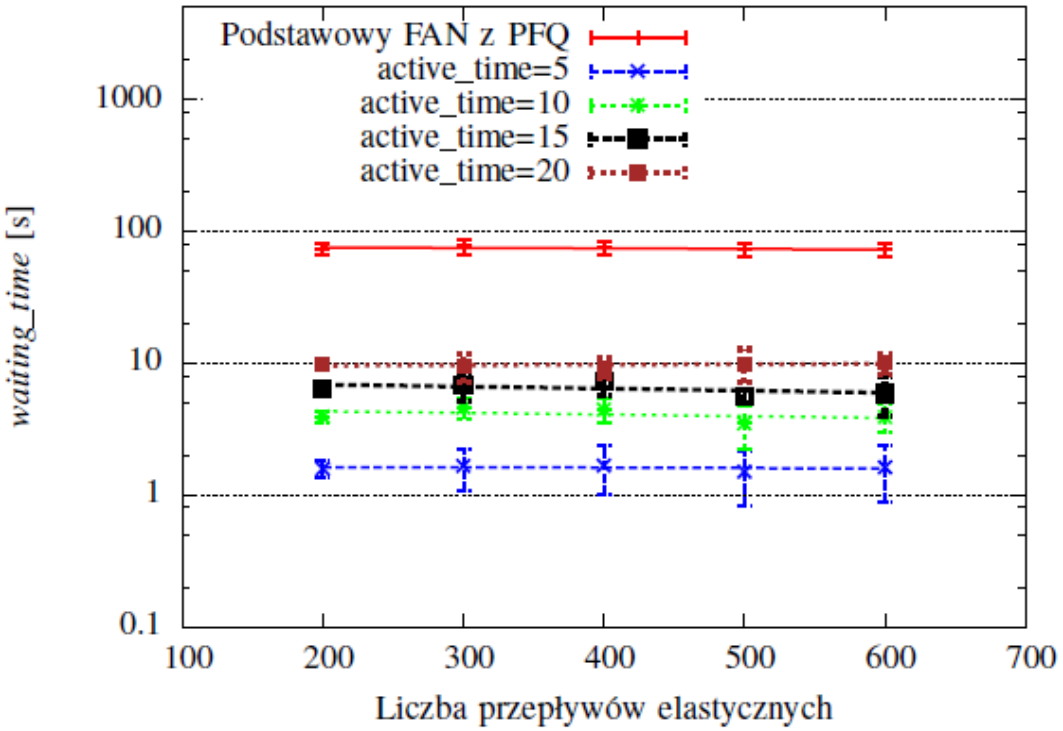
$$y = -91.64 \ln x + 537.54 \quad \text{dla algorytmu PDRR} \tag{11}$$

Korzystając ze wzorów (10) i (11) można wyznaczyć takie wartości parametru *active_time*, dla których nie ulegnie wydłużeniu średni czas transmisji przepływów elastycznych. Są to odpowiednio wartości:

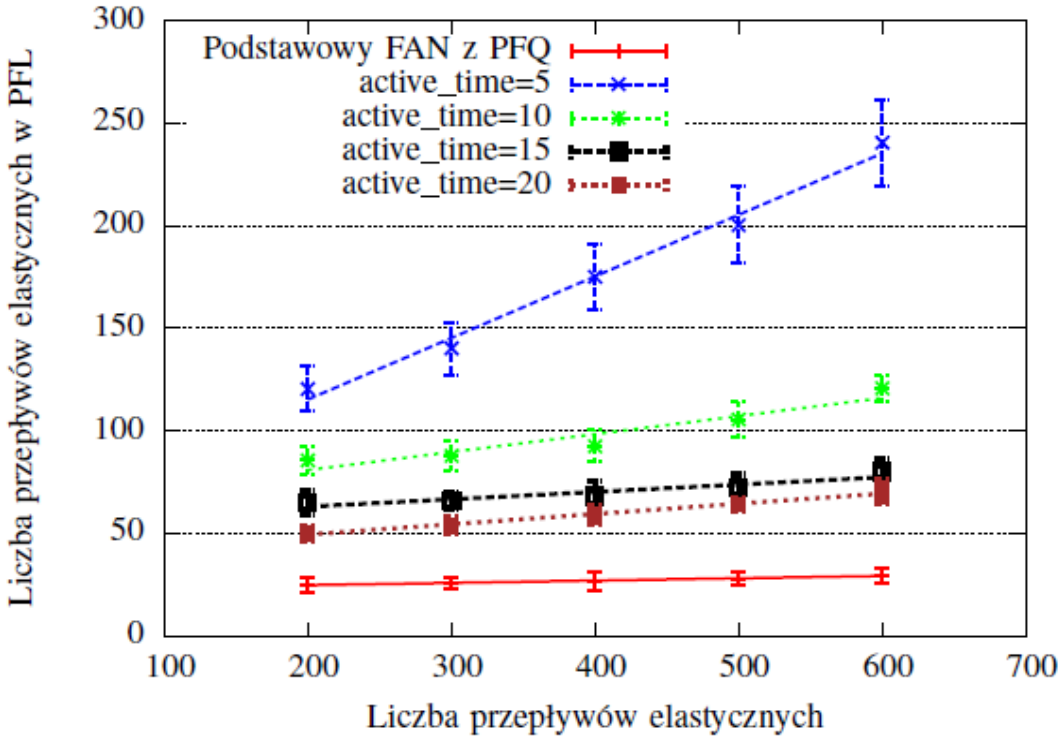
active_time = 33.25 s (dla PFQ)

active_time = 91.45 s (dla PDRR)

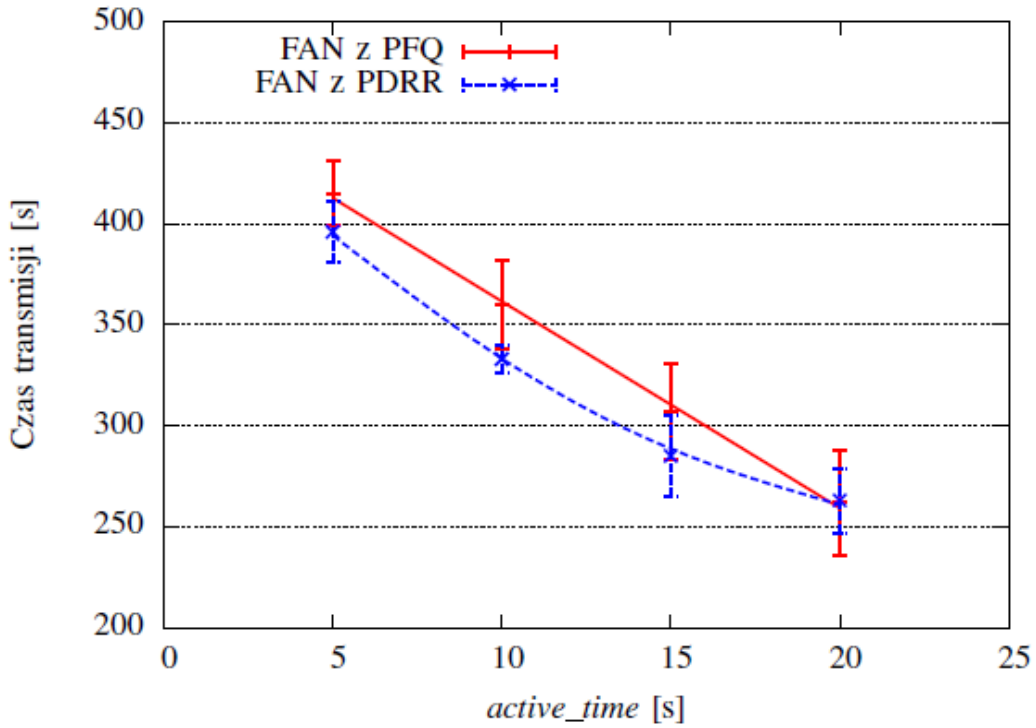
Niestety wartości nie są akceptowalne z punktu widzenia czasu oczekiwania na dostęp do łącza przez przepływy strumieniowe. Są one jednak wyraźnie mniejsze niż w przypadku algorytmu EFM, co wskazuje na lepsze właściwości proponowanego rozwiązania.



Rys. 11. Średni czas akceptacji przepływów strumieniowych w łączu FAN z mechanizmem RAEF



Rys. 12. Liczba przepływów elastycznych w PFL w łączu FAN z mechanizmem RAEF



Rys. 13. Średni czas transmisji przepływów elastycznych w łączu FAN z mechanizmem RAEF

Średni czas akceptacji przepływów strumieniowych w ruterze z zaimplementowanym mechanizmem RBAEF i algorytmem PFQ jest zaprezentowany na Rys. 14. Średnia liczba zaakceptowanych przepływów w bloku sterowania dostępem jest przedstawiona na Rys. 15, a średni czas transmisji przepływów elastycznych w łączu pokazano na Rys. 16.

Uzyskane wyniki wskazują, że odpowiedni czas akceptacji nowych przepływów strumieniowych w łączu można osiągnąć dla *active_time* = 5 s lub 10 s, czyli dla tych samych wartości, co w poprzednich dwóch przypadkach. Niestety, mimo wprowadzenia listy przepływów blokowanych, także i w tym przypadku średnia liczba zaakceptowanych przepływów na liście PFL jest zbyt wysoka w porównaniu z podstawową wersją sieci FAN. Co prawda uzyskane wartości są mniejsze niż we wcześniej analizowanych przypadkach, to jednak wciąż nie są one zadowalające. Niekorzystne wydaje się też blokowanie odrzuconych przepływów z punktu widzenia ich transmisji. Takie działanie, choć w niewielkim stopniu, wydłuża średni czas transmisji przepływów elastycznych. Na Rys. 14 i 15 przedstawiono jedynie wartości dla algorytmu PFQ, gdyż dla PDRR były one podobne.

Krzywe z Rys. 16 można aproksymować poniższymi równaniami:

$y = -108.94 \ln x + 611.43$ dla algorytmu PFQ (12)

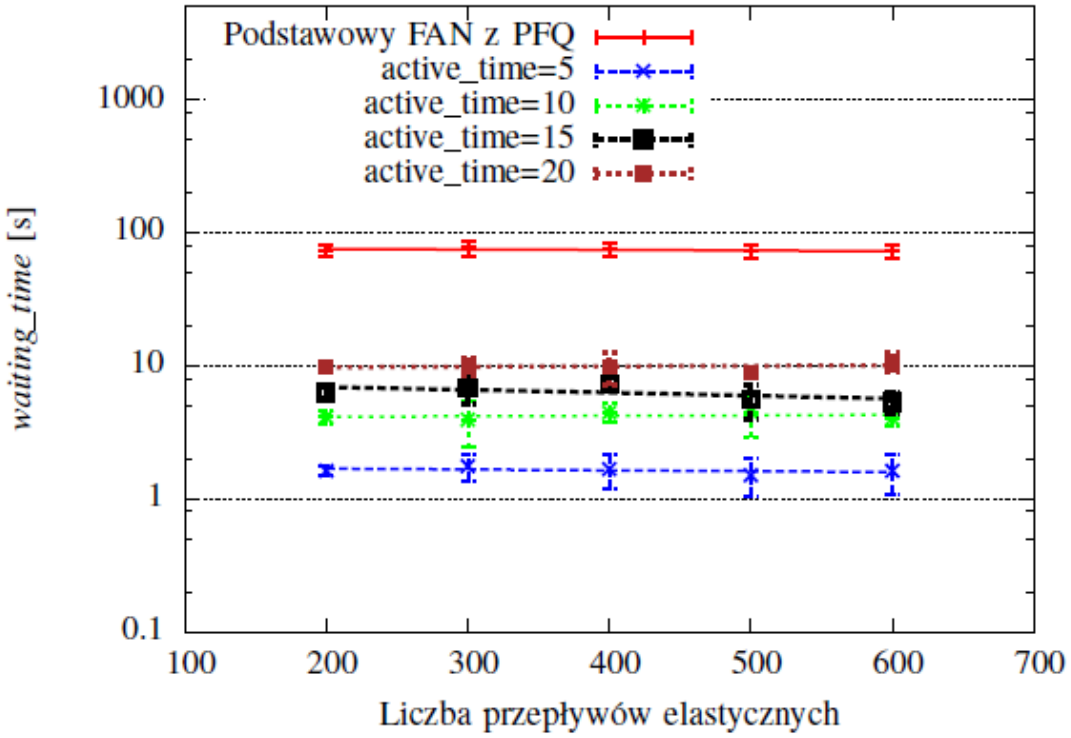
$y = -82.84 \ln x + 521.37$ dla algorytmu PDRR (13)

Korzystając ze wzorów (12) i (13) można wyznaczyć takie wartości parametru *active_time*, dla których nie ulegnie wydłużeniu średni czas transmisji przepływów elastycznych. Są to odpowiednio wartości:

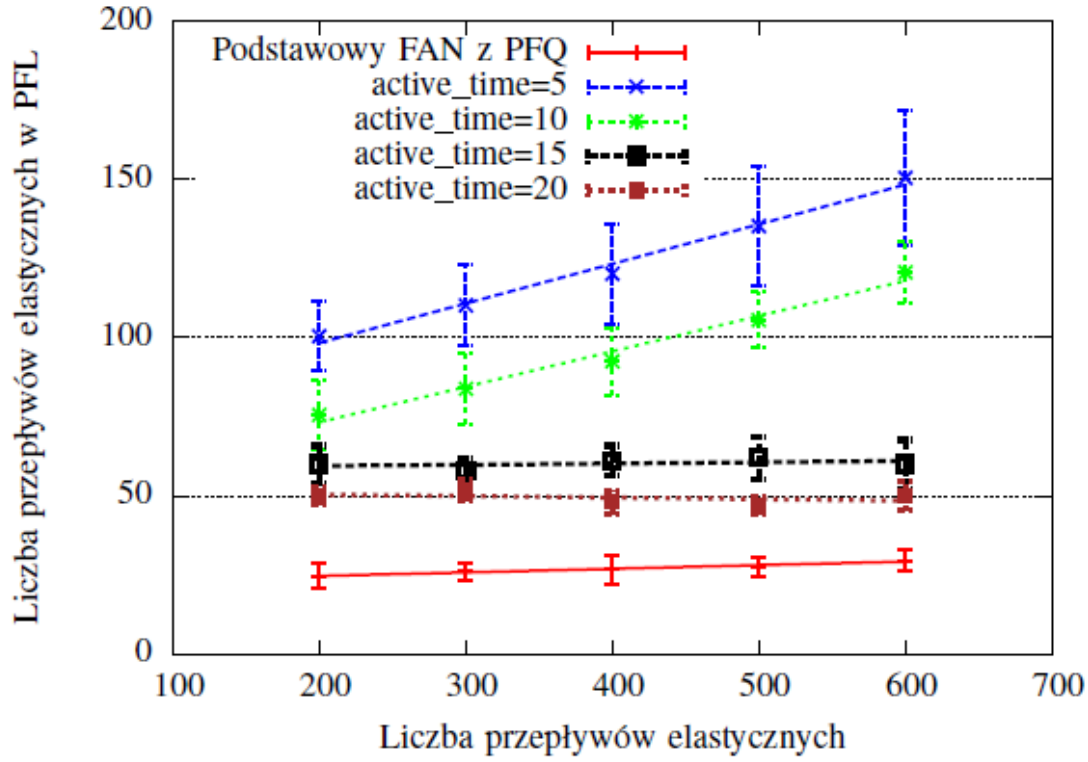
active_time = 35.40 s (dla PFQ)

active_time = 76.40 s (dla PDRR)

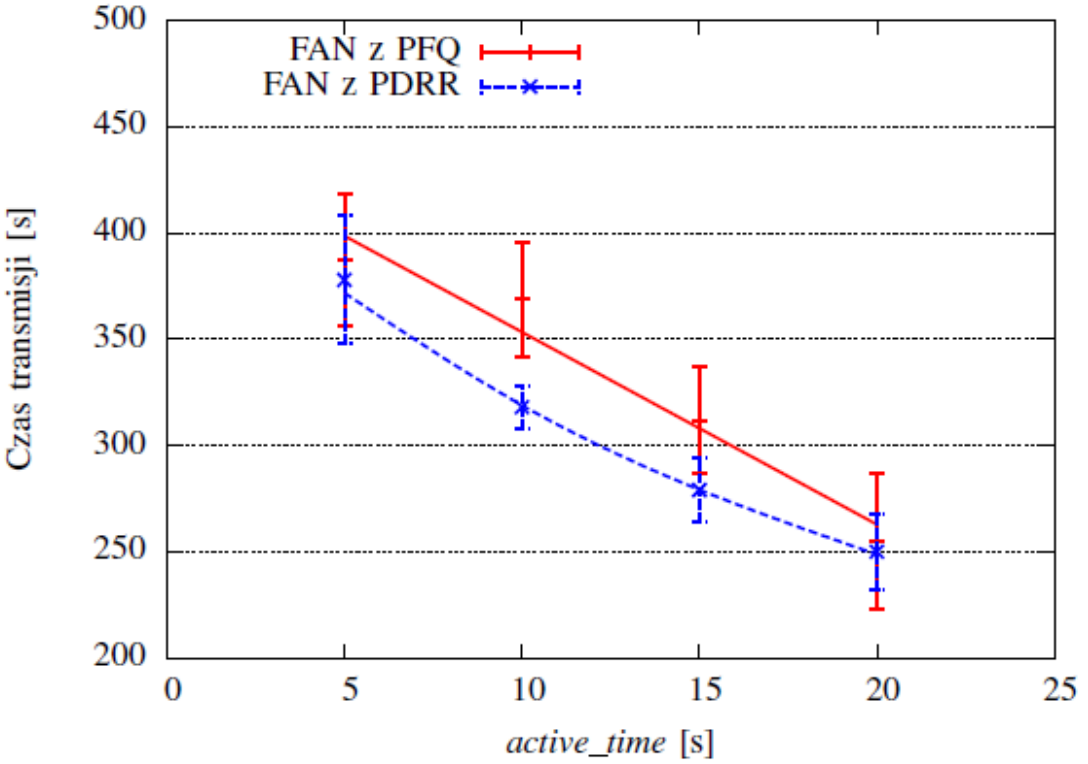
Niestety wartości te nie są akceptowalne z punktu widzenia czasu oczekiwania na dostęp do łącza przez przepływy strumieniowe. Są one jednak znacząco niższe niż w przypadku algorytmu EFM i porównywalne z wartościami uzyskanymi dla algorytmu RAEF.



Rys. 14. Średni czas akceptacji przepływów strumieniowych w łączu FAN z mechanizmem RBAEF



Rys. 15. Liczba przepływów elastycznych w PFL w łączu FAN z mechanizmem RBAEF



Rys. 16. Średni czas transmisji przepływów elastycznych w łączu FAN z mechanizmem RBAEF

Odmienne stanowisko w stosunku do propozycji RBAEF zostało zaprezentowane w mechanizmie RPAEF, gdzie listę przepływów blokowanych zastąpiła lista przepływów z priorytetem w dostępie do łącza. Zabieg ten pozwolił na uzyskanie znacząco lepszych wyników niż w poprzednich propozycjach.

Średni czas akceptacji przepływów strumieniowych w ruterze z zaimplementowanym mechanizmem RPAEF i algorytmem PFQ jest zaprezentowany na Rys. 17, natomiast średnia liczba zaakceptowanych przepływów w bloku sterowania dostępem przedstawiona jest na Rys. 18. Średni czas transmisji przepływów elastycznych w łączu pokazano na Rys. 19.

Uzyskane wyniki wskazują, że zadowalający czas akceptacji nowych przepływów strumieniowych w łączu można osiągnąć dla *active_time* = 5 s lub 10 s (przy czym dla *active_time* = 5 s wyniki są zdecydowanie najlepsze). Średnia liczba zaakceptowanych przepływów na liście PFL po akcji jej czyszczenia jest wciąż większa niż w przypadku podstawowej wersji sieci FAN. Podobnie, czas transmisji przepływów elastycznych ulega znacznemu wydłużeniu dla małych wartości parametru *active_time*. Na Rys. 17 i 18 przedstawiono jedynie wartości dla algorytmu PFQ, gdyż dla PDRR były one podobne.

Wyniki z Rys. 19 można aproksymować poniższymi krzywymi:

$$y = -108.94 \ln x + 611.43 \quad \text{dla algorytmu PFQ} \tag{14}$$

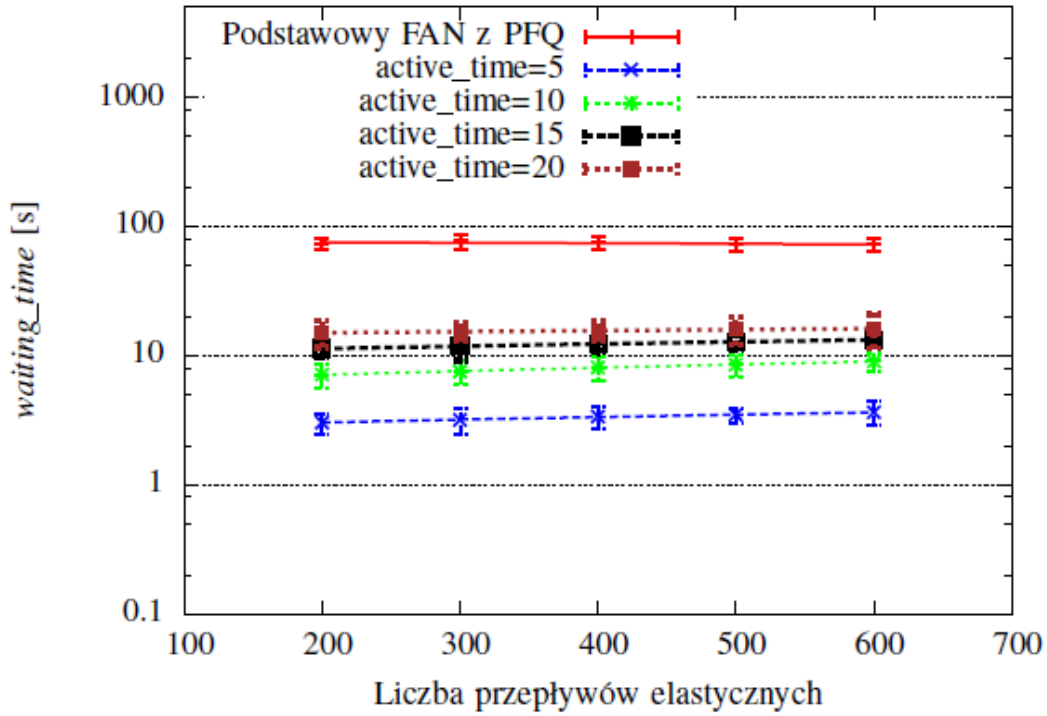
$$y = -82.84 \ln x + 521.37 \quad \text{dla algorytmu PDRR} \tag{15}$$

Korzystając ze wzorów (14) i (15) można wyznaczyć takie wartości parametru *active_time*, dla których nie ulegnie wydłużeniu średni czas transmisji przepływów elastycznych. Są to odpowiednio wartości:

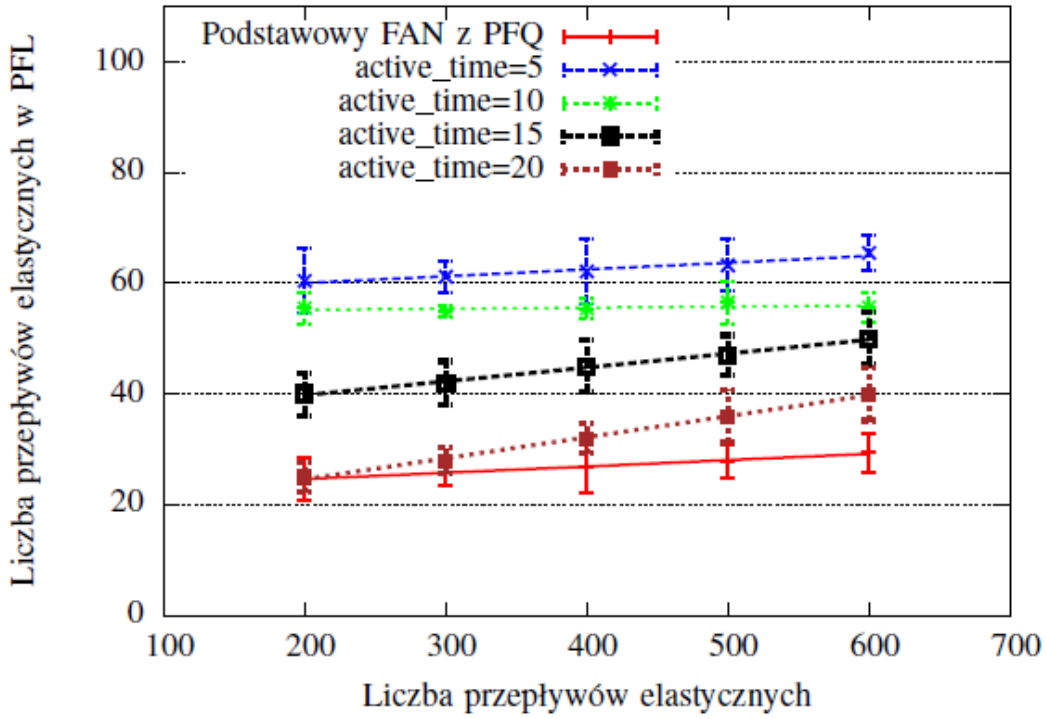
active_time = 20.57 s (dla PFQ)

active_time = 20.40 s (dla PDRR)

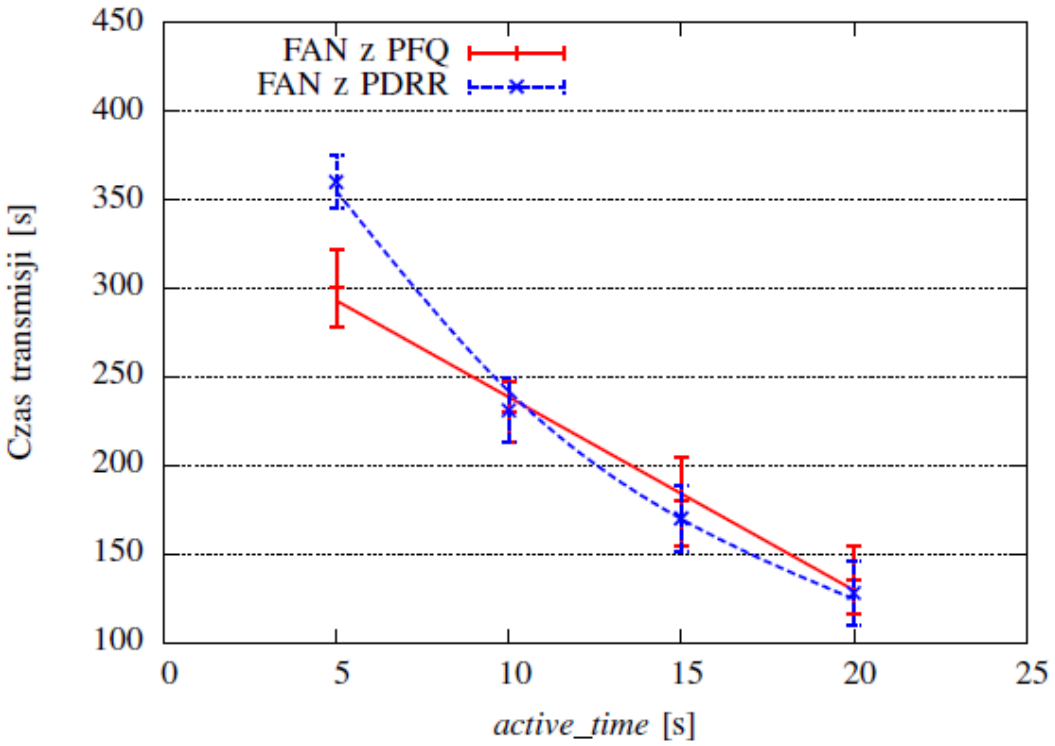
Niestety wartości nie są akceptowalne z punktu widzenia czasu oczekiwania na dostęp do łącza przez przepływy strumieniowe (w obu przypadkach czas oczekiwania przekracza 10 s), jednakże uzyskane wyniki dla mechanizmu RPAEF są zdecydowanie najlepsze spośród wszystkich dotąd prezentowanych.



Rys. 17. Średni czas akceptacji przepływów strumieniowych w łączu FAN z mechanizmem RPAEF



Rys. 18. Liczba przepływów elastycznych w PFL w łączu FAN z mechanizmem RPAEF



Rys. 19. Średni czas transmisji przepływów elastycznych w łączu FAN z mechanizmem RPAEF

Analiza symulacyjna sieci AFAN została przeprowadzona w tej samej topologii i w tych samych warunkach, co w poprzednich przypadkach. Celem takiego działania było umożliwienie porównania otrzymanych wyników.

Średni czas akceptacji przepływów strumieniowych w ruterze zaprojektowanym dla sieci AFAN jest zaprezentowany na Rys. 20, natomiast średnia liczba zaakceptowanych przepływów w bloku sterowania dostępem przedstawiona jest na Rys. 21. Średni czas transmisji przepływów elastycznych w łączu pokazano na Rys. 22.

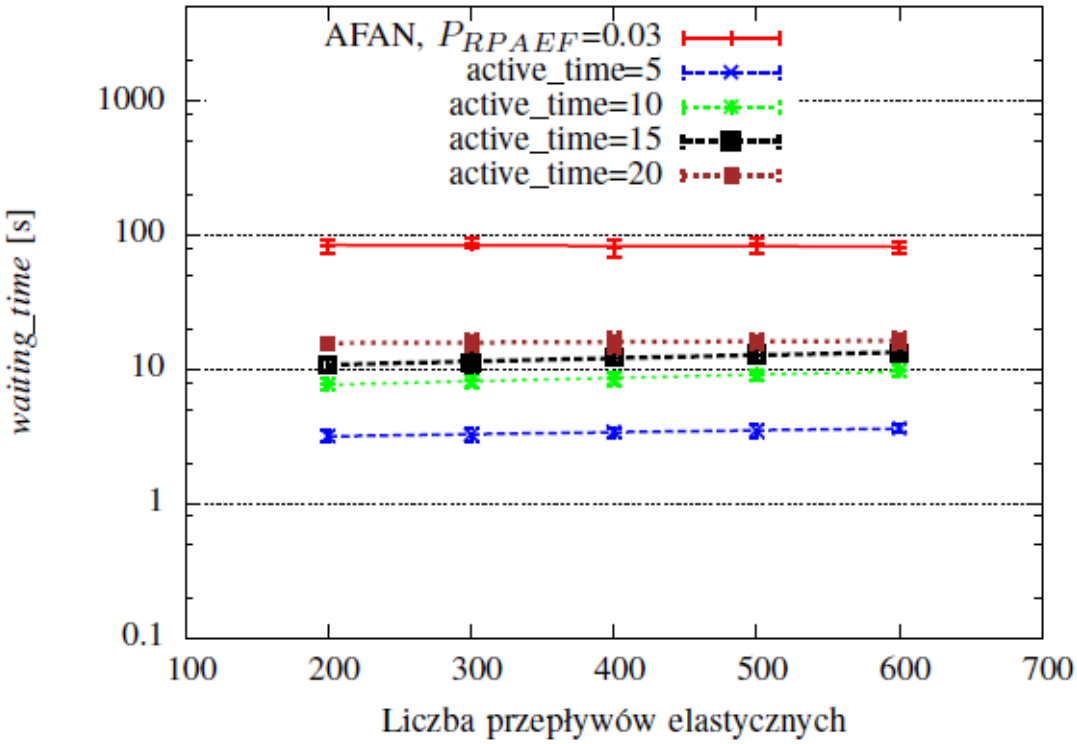
Uzyskane wyniki wskazują, że odpowiedni czas akceptacji nowych przepływów strumieniowych w łączu można osiągnąć dla *active_time* = 5 s. Uzyskana w tym przypadku średnia liczba zaakceptowanych przepływów na liście PFL jest niewiele wyższa niż w przypadku podstawowej wersji sieci FAN. Czas transmisji przepływów elastycznych ulega wydłużeniu dla małych wartości parametru *active_time*, jednak wraz ze wzrostem wartości parametru *active_time* szybko spada.

Krzywą z Rys. 22 (dla P_{RPAEF} = 0.03) można aproksymować poniższym równaniem:

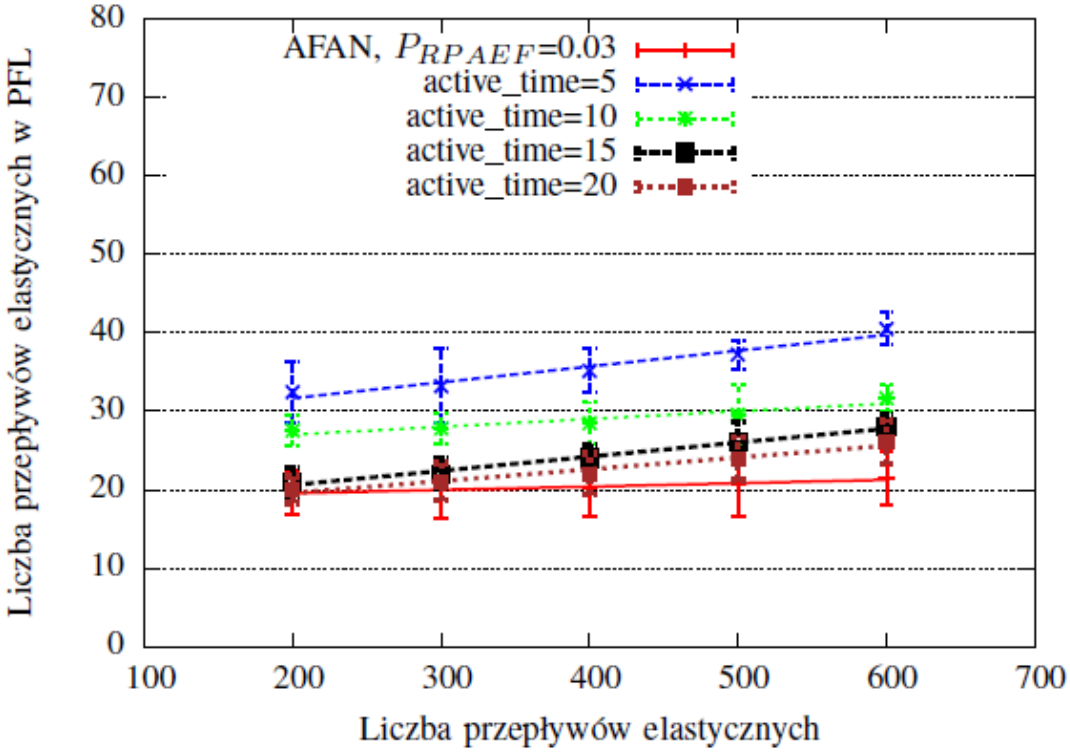
$$y = -17.99 \ln x + 173.96 \quad (16)$$

Wartość P_{RPAEF} = 0.03 została dobrana eksperymentalnie jako najlepsza z możliwych zapewniająca stabilne działanie algorytmu. Korzystając ze wzoru (16) można wyznaczyć taką wartość parametru *active_time*, dla której nie ulegnie wydłużeniu średni czas transmisji przepływów elastycznych. Jest to: *active_time* = 20.77 s.

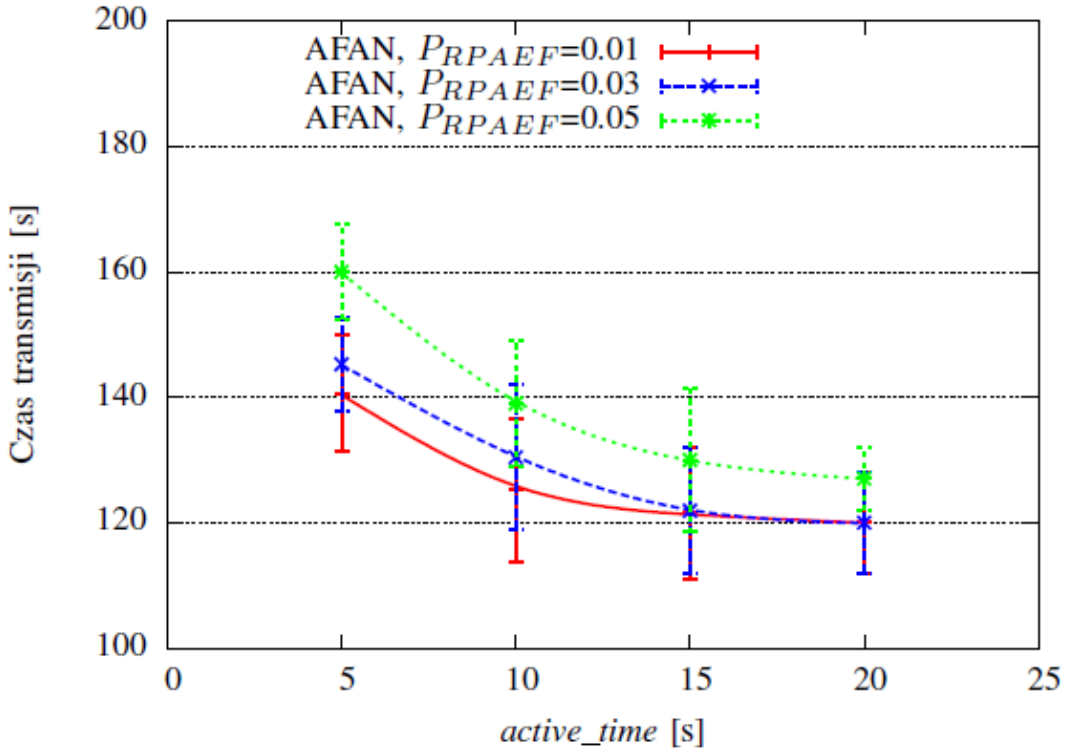
Dla tej wartości parametru *active_time* czasy akceptacji nowych przepływów strumieniowych są wciąż niezadowalające. Jednkaże analiza dla wartośći *active_time* = 8 s pokazuje, że przepływy strumieniowe są akceptowane po średnim czasie 5.62 s ± 0.32 s, a średni czas transmisji przepływów elastycznych wynosi 135.86 s ± 12.82 s. Wartość ta jest porównywalna z analogiczną wartością otrzymaną dla podstawowej wersji FAN. Został więc w tym przypadku spełniony warunek zapewnienia gwarancji dostarczenie szybkiej akceptacji przepływów strumieniowych bez wydłużania czasu transmisji pozostałych przepływów.



Rys. 20. Średni czas akceptacji przepływów strumieniowych w łączu AFAN



Rys. 21. Liczba przepływów elastycznych w PFL w łączu AFAN



Rys. 22. Średni czas transmisji przepływów elastycznych w łączu AFAN

Streszczenie pracy

W rozprawie zaprezentowano mechanizmy sterowania przeciążeniami w sieciach zorientowanych na przepływy (*Flow-Aware Networks*). W szczególności przedstawione zostały mechanizmy EFM (*Enhanced Flushing Mechanism*), RAEF (*Remove and Block Active Elastic Flows*), RBAEF (*Remove and Block Active Elastic Flows*) oraz RPAEF (*Remove and Prioritize in access Active Elastic Flows*) polegające na okresowym, całkowitym bądź częściowym czyszczeniu listy przepływów chronionych w bloku sterowania dostępem. Głównym celem zastosowania mechanizmów sterowania przeciążeniami w sieciach FAN jest zapewnienie możliwie szybkiej akceptacji przepływów strumieniowych w bloku sterowania dostępem. Transmisja strumieniowa w sieciach FAN jest przewidziana do obsługi ruchu o niskiej przepływności z zapewnieniem odpowiednio małych opóźnień i strat pakietów, czyli np. przesyłanie ruchu aplikacji głosowych lub wideo. Istotne jest, by mechanizmy zmniejszające czas rozpoczęcia transmisji dla przepływów strumieniowych nie powodowały znacznego pogorszenia transmisji pozostałego ruchu w sieci. Zaproponowane mechanizmy sterowania przeciążeniami w sieciach FAN zostały szczegółowo opisane i przeanalizowane przy użyciu symulacji przeprowadzonych w symulatorze ns-2. Uzyskane wyniki pozwalają wnioskować, ze nowe rozwiązania umożliwiają znaczące zmniejszenie czasów akceptacji dla nowych przepływów strumieniowych przy jednoczesnym braku zmian czasu transmisji przepływów elastycznych. Co więcej, możliwe jest zapewnienie krótkich czasów akceptacji dla nowych przepływów strumieniowych, spełniających wymagania dla strumieni realizujących transmisje głosowe (w szczególności rozmowy typu VoIP).

Druga część rozprawy stanowi nowa propozycja realizacji koncepcji Flow- Aware Networking. W rozwiązaniu tym zastosowano algorytm losowego usuwania pakietu z kolejki w sytuacji wystąpienia natłoku, zaimplementowany z użyciem mechanizmu AFD (*Approximate Fair Dropping*). Nowa propozycja jest prostsza w implementacji i pozwala na uzyskiwanie wyników porównywalnych z innymi rozwiązaniami sieci FAN.

Wnioski końcowe

Pierwszym celem rozprawy było zaproponowanie nowych mechanizmów sterowania przeciążeniami dla sieci FAN, a mianowicie: EFM (*Enhanced Flushing Mechanism*), RAEF (*Remove Active Elastic Flows*), RBAEF (*Remove and Block Active Elastic Flows*) oraz RPAEF (*Remove and Prioritize in access Active Elastic Flows*). Mechanizmy te, oparte na okresowym całkowitym bądź częściowym czyszczeniu zawartości listy przepływów chronionych PFL w sytuacji przeciążenia łącza, zostały przeanalizowane teoretycznie oraz zbadane przy użyciu symulatora sieciowego ns-2.

W ruterze wzajemnie zabezpieczonym bez dodatkowych mechanizmów nie jest możliwe zaakceptowanie nowych przepływów w sytuacji natłoku. Może to spowodować, że przepływy strumieniowe, takie jak rozmowy VoIP będę musiały zbyt długo oczekiwać na akceptację. Zaproponowane w rozprawie mechanizmy stanowią odpowiedź na tę niedogodność. Dzięki ich zastosowaniu (z odpowiednio dobranymi parametrami), czasy akceptacji nowych przepływów strumieniowych ulegają znacznemu skróceniu. Najlepsze właściwości spośród zaproponowanych rozwiązań posiada RPAEF. W rozwiązaniu tym udało się w znacznym stopniu ograniczyć wydłużenie czasu transmisji przepływów elastycznych. Ponadto mechanizm RPAEF posiada dobre właściwości, gdy chodzi o skalowalność sieci.

Drugim ważnym osiągnięciem opisanym w doktoracie jest nowa propozycja architektury rutera wzajemnie zabezpieczonego – AFAN (Approximate FAN). Dzięki zastosowaniu mechanizmu RED oraz algorytmu AFD zapewniono funkcjonalność sieci FAN w nowej architekturze rutera. Analiza teoretyczna i symulacyjna nowego rozwiązania wskazuje, że jest ono mniej złożone obliczeniowo i łatwiejsze w implementacji. Co więcej, zastosowanie mechanizmu RPAEF pozwala nie tylko zredukować czas akceptacji dla przepływów strumieniowych do akceptowalnego poziomu, ale także nie zwiększać przy tym średniego czasu transmisji przepływów elastycznych.

Najważniejsze osiągnięcia rozprawy przedstawiają się następująco:

- Przedstawiona została dogłębna analiza sieci FAN – wskazane zostały ich zalety i wady.
- Zaprezentowane zostały nowe metody obliczania parametrów *priority_Load* oraz *fair_rate* – pozwalają one na bardziej stabilną transmisję w sieciach FAN i są bardziej odpowiednie do analizy wydajności sieci.
- Przedstawione i przeanalizowane zostały cztery mechanizmy sterowania przeciążeniami (EFM, RAEF, RBAEF oraz RPAEF) – badania symulacyjne zostały przeprowadzone w dwóch różnych architekturach rutera wzajemnie zabezpieczonego (z PFQ lub PDRR).
- Przedstawione i przeanalizowane zostało również działanie mechanizmów sterowania przeciążeniami w sieci w sytuacji wystąpienia uszkodzenia łącza. Zaproponowane mechanizmy i w tym przypadku pozwalają na znaczącą poprawę jakości transmisji w sieciach FAN.
- Nowa wersja rutera wzajemnie zabezpieczonego (AFAN) została zaprezentowana w drugiej części rozprawy. Badania symulacyjne potwierdziły, że rozwiązanie to jest bardziej wydajne przy zachowaniu podobnych parametrów transmisji co w przypadku rutera z algorytmem PFQ lub PDRR.
- Przeprowadzona analiza symulacyjna pokazała, że również i dla tej wersji rutera wzajemnie zabezpieczonego, algorytmy sterowania przeciążeniami znacząco usprawniają pracę sieci.

- Zastosowanie algorytmu RPAEF w architekturze AFAN pozwoliło na osiągnięcie najlepszych, w pełni akceptowalnych wyników.

Wyżej wskazane osiągnięcia pokazują jednoznacznie, że teza pracy została udowodniona.

Literatura

[1] J. Domzal and A. Jajszczyk. New Congestion Control Mechanisms for Flow-Aware Networks. In *Proceedings of International Conference on Communications, ICC 2008*, Beijing, China, May 2008.

[2] J. Domzal and A. Jajszczyk. The Flushing Mechanism for MBAC in Flow-Aware Networks. In *Proceedings of 4th EURO-NGI Conference on Next Generation Internet Networks, NGI 2008*, pages 77–83, Krakow, Poland, April 2008.

[3] J. Domzal and A. Jajszczyk. The Impact of Congestion Control Mechanisms for Flow-Aware Networks on Traffic Assignment in Two Router Architectures. In *Proceedings of International Conference on the Latest Advances in Networks, ICLAN 2008*, Toulouse, France, December 2008.

[4] J. Domzal and A. Jajszczyk. The Impact of Congestion Control Mechanisms on Network Performance after Failure in Flow-Aware Networks. In *Proceedings of International Workshop on Traffic Management and Traffic Engineering for the Future Internet, FITraMEn 2008*, Porto, Portugal, December 2008.

[5] ITU-T Recommendation E.721. *Network grade of service parameters and target values for circuit-switched services in the evolving ISDN*, May 1999.

[6] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. On the scalability of fair queuing. In *Proceedings of Third Workshop on Hot Topics in Networks, ACM HotNets-III 2004*, San Diego, USA, November 2004.

[7] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing. In *Proceedings of International Conference on Measurement and modeling of computer systems, ACM SIGMETRICS 2005*, Banff, Canada, June 2005.

[8] A. Kortebi, S. Oueslati, and J. Roberts. Cross-protect: implicit service differentiation and admission control. In *Proceeding of High Performance Switching and Routing, HPSR 2004*, Phoenix, USA, April 2004.

[9] A. Kortebi, S. Oueslati, and J. Roberts. Implicit Service Differentiation using Deficit Round Robin. In *Proceedings of 19th International Teletraffic Congress, ITC19*, Beijing, China, August/September 2005.

[10] The Network Simulator ver. 2. Available at http://www.isi.edu/nsnam/ns/.

[11] S. Oueslati and J. Roberts. Method and a device for implicit differentiation of quality of service in a network. *United States Patent 2004/0213265 A1*, Oct. 2004.

[12] J. Roberts, S. Oueslati, and A. Kortebi. Method and device for scheduling packets for routing in a network with implicit determination of packets to be treated as priority. *United States Patent 2007/0291644 A1*, Dec. 2007.

[13] J. Roberts. Internet Traffic, QoS and Pricing. *Proceedings of the IEEE*, 92:1389–1399, September 2004.

[14] J. Roberts and S. Oueslati-Boulahia. Quality of Service by Flow Aware Networking. *Philosophical Transactions of Royal Society*, 358(1773):2197–2207, Aug. 2000.