Algorytmy kolejkowania. Algorytm RED (Random Early Detection)

Queueing algorithms. Algorithm RED (Random Early Detection)

Paweł Iljaszewicz[1]

STRESZCZENIE: Artykuł omawia algorytm Losowego Wczesnego Wykrywania RED (*ang. Random Early Detection*) pozwalający bramce unikania przeciążeń w sieciach z komutacją pakietów. Brama wykrywa początkowe przeciążenie, obliczając średni rozmiar kolejki. Brama może powiadamiać o przeciążonych połączeniach lub o upuszczeniu pakietów przybywających do bramy, ustawiając bit w nagłówkach pakietów. Kiedy rozmiar średniej kolejki przekracza ustawiony próg, brama opada lub zaznacza każdy przybywający pakiet z pewnym prawdopodobieństwem, gdzie dokładny rozkład prawdopodobieństwa jest funkcją średniego rozmiaru kolejki. Bramki RED utrzymują średnią wielkość kolejki na niskim poziomie, jednocześnie zezwalając na sporadyczne impulsy pakietów w kolejce. Podczas przeciążenia prawdopodobieństwo, że brama powiadamia o konkretnym połączeniu, by zmniejszyć jego okno, jest mniej więcej proporcjonalne do udziału tego w przepustowości przez bramę. Bramki RED są zaprojektowane tak, aby dostarczyć protokół taki jak TCP, przeciążając warstwę transportową. Symulacje sieci TCP / IP są używane do zilustrowania wydajności bramki.

SŁOWA KLUCZOWE: Aktywne zarządzanie kolejkami (AQM), metody kolejkowania, algorytm RED, przepustowość sieci.

ABSTRACT: The subject of the study is to present the Random Early Detection (RED) algorithm that allows the gateway to avoid overloading in packet switched networks. The gateway detects the initial overload by calculating the average size of the queue. The gateway can notify about overload connections or by dropping packets arriving at the gate by setting a bit in the packet headers.
When the size of the average queue exceeds the set threshold, the gate descends or marks each arriving packet with a certain probability, where the exact probability distribution is a function of the average queue size.
RED gates maintain the average queue size at a low level, while allowing occasional packet bursts in the queue. During overload, probability that the gateway informs about a specific connection to reduce its window is more or less proportional to this connection involved in bandwidth through the gate. The RED gateways are designed to provide a protocol such as TCP to overload the transport layer. TCP / IP network simulations are used to illustrate the performance of the gateway.

KEYWORDS: Active queue management (AQM), queuing methods, RED algorithm, network bandwidth.

1. Wprowadzenie

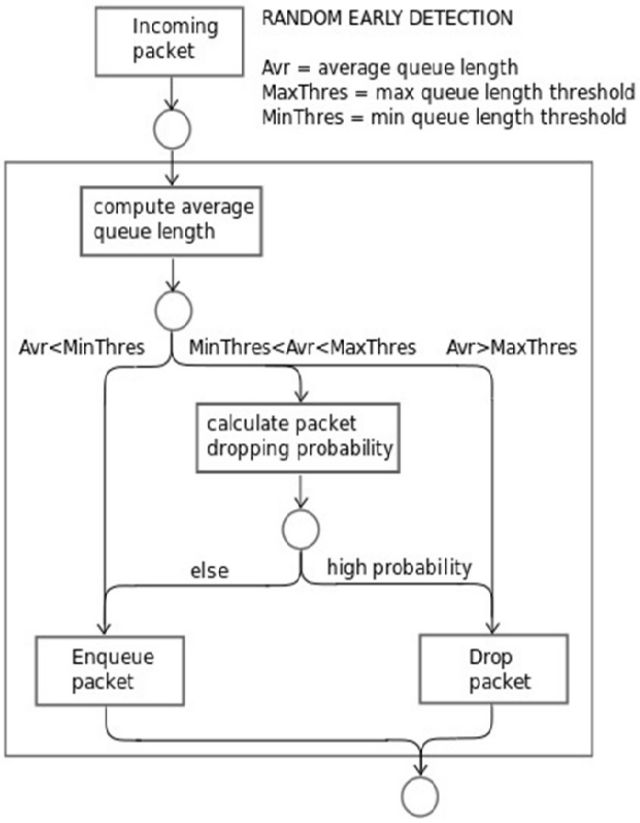
W przeciwieństwie do sieci domowych, problem przepustowości dla szybkobieżnych sieci z połączeniami powyżej 100 megabitów wymaga odpowiedniego projektowania. Aby uniknąć przejściowego przeciążenia, kolejkuje się przesyłanie danych, co powoduje wysyłanie ich z dużym opóźnieniem. W internecie protokół transportowy TCP wykrywa przeciążenie tylko po pakiecie przychodzącym do bramy. Taka implementacja dużych kolejek jest niepożądana, gdyż duże kolejki (opóźnienie wynikające z przepustowości produktu) powodują znaczne zwiększenie średniego opóźnienia sieci. Dlatego przy coraz szybszych sieciach coraz ważniejsze jest posiadanie mechanizmów, które utrzymują przepustowość na wysokim poziomie, przy zachowaniu średniej wielkości kolejki na niskim. [1, 2]
Standardowym działaniem jest odrzucanie nadmiaru (*ang. Tail-drop*). Polega ono na przyjmowaniu określonej liczby pakietów do ustalonej (zadanej) wielkości, a wszystkie pozostałe są przekierowane na inną linię lub czekają na retransmisję, czyli powtórne wysłanie. Taka sytuacja po-

1. Instytut Technologiczno-Przyrodniczy, Falenty, al. Hrabska 3, 05-090 Raszyn. Email: p.iljaszewicz@itp.edu.pl

legająca na odrzuceniu serii pakietów, powoduje kolejne zapychanie się bramy. Aby temu zapobiec, tworzy się kolejki, co prawda powoduje to zwiększenie przepustowości, ale zwiększają się opóźnienia w wysyłaniu pakietów.
Jednym z rozwiązań jest zastosowanie algorytmu RED, (skrót od Random Early Detect), zwanego także Random Early Drop, co można przetłumaczyć na algorytm Losowego Wczesnego Wykrywania lub algorytm Losowego Wczesnego Odrzucania.[3, 4, 5]

2. Ogólny Algorytm RED

Na rysunku 1 przedstawiono blokowy schemat działania algorytmu RED.[6,7]
RED monitoruje średni rozmiar kolejki Avr (*ang.average queue length*) i przepuszcza (lub znakuje, gdy używane w połączeniu z ECN[2]) pakiety na podstawie statystycznych prawdopodobieństw. Jeśli bufor jest prawie pusty, wówczas wszystkie przychodzące pakiety są akceptowane. W miarę wzrostu kolejki rośnie również prawdopodobieństwo upuszczenia przychodzącego pakietu. Gdy bufor jest pełny, prawdopodobieństwo osiągnęło 1, a wszystkie przychodzące pakiety zostały odrzucone.
RED jest bardziej sprawiedliwy niż odrzucanie nadmiaru (*ang. Tail-drop*), w tym sensie, że nie ma tendencyjności przeciwko ruchowi seryjnemu (*ang. Bursty*), który wykorzystuje tylko niewielką część przepustowości. Im więcej host transmituje, tym bardziej prawdopodobne jest to, że jego pakiety są odrzucane, ponieważ prawdopodobieństwo, że pakiet hosta zostanie odrzucony, jest proporcjonalne do ilości danych, które ma w kolejce. Wczesne wykrywanie pomaga uniknąć globalnej synchronizacji TCP. MaxThres i MinThres oznacza odpowiednio maksymalny i minimalny próg długości kolejki pakietów. MinThres określa minimalny próg kolejki, od którego rozpocznie się odrzucanie. MaxThres to górna granica, której algorytm nie przekroczy, a seria to maksymalna liczba pakietów, która może zostać wysłana jednorazowo.
Ogólny algorytm RED bramki jest pokazany na rysunku 2.
Bramka RED oblicza średni rozmiar kolejki za pomocą filtra dolnoprzepustowego z wykładniczą ważoną średnią kroczącą. Średni rozmiar kolejki jest porównywany z dwoma progami: minimalnym i maksymalnym. Kiedy średni rozmiar kolejki jest mniejszy niż minimalny próg, żadne pakiety nie są oznaczone. Gdy średni rozmiar kolejki jest większy niż próg maksymalny, każdy pakujący pakiet jest oznaczony. Jeśli zaznaczone pakiety są w rzeczywistości zrzucone lub wszystkie węzły źródłowe są kooperatywne, zapewnia to nam, że średni rozmiar kolejki nie przekracza znacząco maksymalnego progu.
Gdy średni rozmiar kolejki jest w zakresie od minimalnego do maksymalnego progu, każdy przybywający pakiet jest oznaczony symbolem prawdopodobieństwa p_a, gdzie p jest funkcją średniej kolejki avg.
Za każdym razem gdy jest zaznaczony pakiet, prawdopo



Ryc. 1 Schemat działania algorytmu RED[3]
Fig. 1 RED principle of operation diagram

dobieństwo, że pakiet jest oznaczony z określonego połączenia, jest proporcjonalne do udziału tego łącza w przepustowości na poziomie bramki.

```
for each packet arrival
   calculate the average queue size avg
   if min_th ≤ avg < max_th
      calculate probability p_a
      with probability p_a:
         mark the arriving packet
   else if max_th ≤ avg
      mark the arriving packet
```

Ryc. 2 Schemat algorytmu RED
Fig. 2 Pseudocode of RED algorithm

Zatem bramka RED ma dwa oddzielne algorytmy. Algorytm obliczania średniej wielkości kolejki określa stopień wielkości serii, który będzie dozwolony w kolejce bramy. Algorytm do obliczania prawdopodobieństwa oznaczania pakietów określa częstotliwość podawania znaczników bramek dla bieżącego poziomu przeciążenia. Celem jest brama znakująca paczki w dość równomiernie rozmieszczonych odstępach, aby uniknąć uprzedzenia i unikanie globalnej synchronizacji oraz zaznaczanie pakietów wystarczająco często, aby kontrolować średni rozmiar kolejki.

2. Explicit Congestion Notification Jawne Powiadomienie o Zatorach rozszerzenie protokołu TCP
3. http://www.icir.org/floyd/red.html

3. Szczegółowy algorytm dla bramek RED

Na rysunku 3 przedstawiono dokładny algorytm dla bramek RED [5]. Widzimy, że algorytm bramki RED można efektywnie wdrożyć z niewielką liczbą instrukcji dodawania i zmian dla każdego pakietu przychodzącego. Ponadto algorytm bramki RED nie jest ściśle sprzężony z przekazywaniem pakietów, a jego obliczenia nie muszą być wykonywane w krytycznej czasowo ścieżce przekazywania pakietów. Znaczna część pracy algorytmu bramki RED taka jak obliczenie średniego rozmiaru kolejki i prawdopodobieństwa oznaczania pakietów p_b może być wykonywane równolegle z przekazywaniem pakietów lub może być obliczona przez bramę jako zadanie o niższej liczbie priorytetów, jeśli pozwala na to czas. Oznacza to, że algorytm bramki RED nie musi zakłócać działania bramy i umiejętności przetwarzania pakietów, a algorytm bramki RED może być dostosowany do coraz szybszego wyjścia pakietów.

```
Initialization:
    avg ← 0
    count ← −1
for each packet arrival
    calculate the new average queue size avg:
        if the queue is nonempty
            avg ← (1 − w_q)avg + w_q q
        else
            m ← f(time − q_time)
            avg ← (1 − w_q)^m avg
    if min_th ≤ avg < max_th
        increment count
        calculate probability p_a:
            p_b ← max_p(avg − min_th)/(max_th − min_th)
            p_a ← p_b/(1 − count · p_b)
        with probability p_a:
            mark the arriving packet
            count ← 0
    else if max_th ≤ avg
        mark the arriving packet
        count ← 0
    else count ← −1
when queue becomes empty
    q_time ← time

Saved Variables:
    avg:   average queue size
    q_time:   start of the queue idle time
    count:   packets since last marked packet

Fixed parameters:
    w_q:   queue weight
    min_th:   minimum threshold for queue
    max_th:   maximum threshold for queue
    max_p:   maximum value for p_b

Other:
    p_a:   current packet-marking probability
    q:   current queue size
    time:   current time
    f(t):   a linear function of the time t
```

Ryc. 3 Dokładny algorytm dla bramek RED

Jeśli metoda znakowania pakietów przez bramę RED ma ustawić bit wskazania przeciążenia w nagłówku pakietu, zamiast upuszczania nadchodzącego pakietu, ustawienie samego wskaźnika sygnalizującego zator pakietów dodaje do niego znacznik algorytmu bramy. Jednakże, ponieważ bramki RED są zaprojektowane do oznaczania jak najmniejszej liczby pakietów, narzut ustawienia bitów sygnalizacji przeciążenia jest ograniczony do minimum.
W przeciwieństwie do bram DECbit[4], który ustawia bit wskazania przeciążenia w każdym pakiecie, który dociera do bramy, gdy średni rozmiar kolejki przekracza próg. Dla każdego przybycia pakietu do kolejki bramy, bramka RED oblicza średni rozmiar kolejki. To można przedstawić w następujący sposób:

$$avg \leftarrow avg + wq(q - avg)$$

Tak długo jak wq (*ang. queue weight*) kolejka zadań jest wybierana jako (ujemna) potęga dwóch, można to zrealizować za pomocą jednej zmiany i dwóch dodatków (ze skalowanymi wersjami parametrów) [7].
Ponieważ bramka RED oblicza średni rozmiar kolejki przy odbiorze pakietu, a nie w ustalonym czasie interwału, obliczanie średniego rozmiaru kolejki jest modyfikowane, gdy pakiet dociera do bramy do pustej kolejki. Gdy pakiet dotrze do bramy do pustej kolejki, bramka oblicza m liczbę pakietów, które mogły zostać przesłane przez bramę w czasie, kiedy linia była wolna.
Bramka oblicza średni rozmiar kolejki tak, jakby *m* przybyło do bramy z rozmiarem kolejki zero. Obliczenia są następujące:

$$(1-wq)^{(time-q\ time)/s}$$

$$avg \leftarrow (1-wq)^m avg$$

Gdzie *q*_time jest początkiem czasu bezczynności kolejki, a *s* jest typowym czasem transmisji małego pakietu. To całe obliczenie jest przybliżeniem, ponieważ jest oparte na liczbie pakietów, które mogły dotrzeć do bramy przez pewien czas. Po obliczeniu czasu bezczynności ($time - q\_time$) ma przybliżoną wartość poziomu dokładności, można użyć wyszukiwania tabeli, aby uzyskać warunek, $(1-wq)^{(time-q\ time)/s}$, który powinien w przybliżeniu być potęgą liczby 2.

4. Implementacja algorytmu RED

Gdy pakiet dotrze do bramy, a średni rozmiar kolejki *avg* przekracza próg maksymalny max_{th}, to taki przybywający pakiet jest oznaczany. Nie ma ponownego przeliczania prawdopodobieństwa oznaczania pakietów. Jednakże, gdy pakiet a dociera do bramy, a średnia wielkość kolejki *avg* między dwoma progami to min_{th} i max_{th}, prawdopodobieństwo p_b oznaczenia początkowego pakietu oblicza się w następujący sposób:

$$p_b \leftarrow C_1 avg - C_2$$

4. DECbit- Mechanizm decyzyjny ograniczania ruchu - został opracowany do użytku w architekturze cyfrowej sieci (Digital Network Architecture DNA), sieci bezpołączeniowej zawierającej protokół transportowy.

Dla:

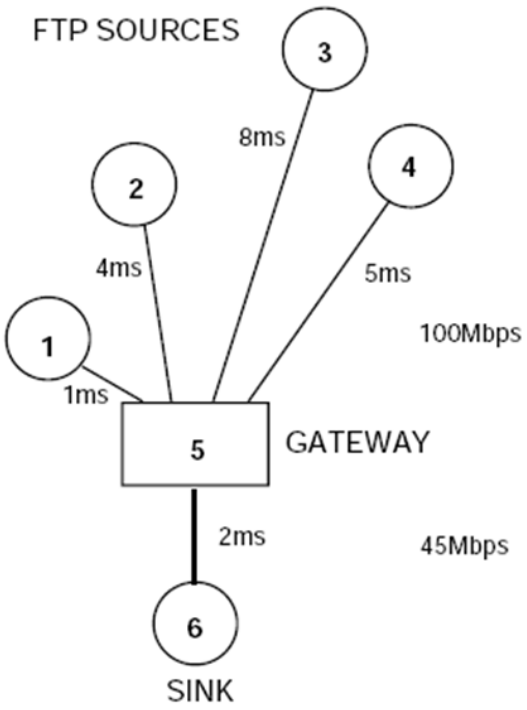
$$C_1 = \frac{max_p}{max_{th} - min_{th}}$$

$$C_2 = \frac{max_p min_{th}}{max_{th} - min_{th}}$$

Parametry max_p, max_{th} i min_{th} są ustalonymi parametrami określonymi wcześniej. Wartości dla max_{th} i min_{th} są określane przez pożądane ograniczenia średniego rozmiaru kolejki i mogą być ograniczone w sposób elastyczny. Jednak ustalony parametr max_p można łatwo ustawić na zakres wartości. W szczególności max_p można wybrać tak, aby C_1 był potęgą dwóch. W ten sposób obliczenie p_b można osiągnąć za pomocą jednej zmiennej z jedną instrukcją dodawania.

5. Obliczenia i analiza wyników

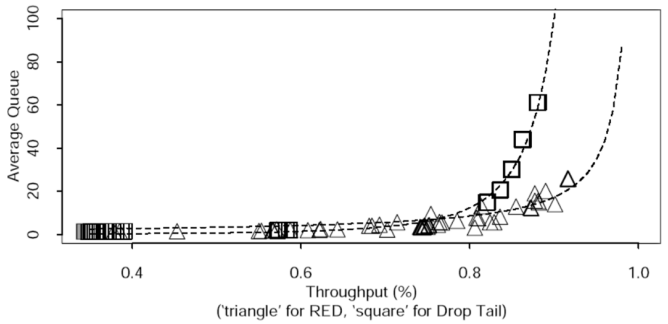
W pracach [8, 12] przedstawiono badanie symulacyjne algorytmu RED. Na rysunku 4 widzimy schemat symulowanej sieci i rozmieszczenie węzłów.



Ryc. 4 Schemat symulowanej sieci
Fig. 4 Diagram of simulated network

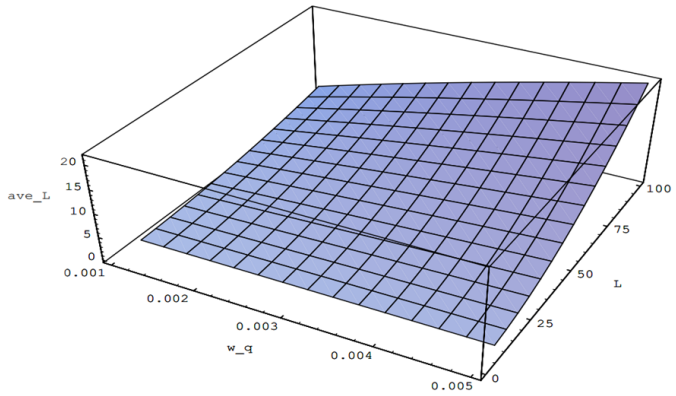
Na rysunku 5 porównano przepustowość pakietów na bramkach Drop Tail i RED.
Symulacje z bramkami RED były uruchamiane z rozmiarem bufora 100 pakietów, z dziewiątym zakresem od 3 do 50 pakietów. Dla bramek RED wartość max_{th} jest ustawiona na $3min_{th}$, z $wq = 0{,}002$ i $max_p = 1/50$. Linie przerywane pokazują średnie opóźnienie (w funkcji przepustowości) przybliżone o $1{,}73/(1- x)$ dla symulacji z bramkami RED i przybliżone przez $0{,}1 = (1-x)^3$ dla symulacji bramy z Drop Tail.

Węzeł 1 zaczyna przesyłać w czasie 0,0 sek.
Węzeł 2 zaczyna przesyłać w czasie 0,2 sekundy
Węzeł 3 zaczyna przesyłać w czasie 0,4 s
Węzeł 4 zaczyna przesyłać w czasie 0,6 sekundy



Ryc. 5 Porównanie bram Drop Tail (kwadraty) i RED (trójkąty)
Fig. 5 Comparison of gateways Drop Tail (square) and RED (triangle)

Autorzy [12, 13] przypisują ulepszoną przepustowość do zredukowanej średniej długości kolejki. Pakiety oczekują mniej czasu w kolejce i szybciej się rozłączają.
Można porównać to do szybkiej jazdy samochodem. Wyprzedzanie większej liczby samochodów nie oznacza, że większa ilość pojazdów na jednostkę czasu dotrze do celu.



Ryc. 6 Graficzne przedstawienie zależności pomiędzy *avg* a *wq*
Fig 6. Graphical representation of relation between avg and wq

Jeśli kolejka zadań *wq* jest zbyt długa, wówczas procedura uśredniania nie odfiltruje chwilowego przeciążenia w bramie. Załóżmy, że kolejka jest początkowo pusta, ze średnim rozmiarem kolejki równym zero, a następnie kolejka wzrasta od 0 do *L* pakietów przez *L* pakietów przybycia. Po przybyciu pakietu *L* - tego do bramy, średnia kolejka ma rozmiar avg_L dany wzorem[8]:

$$avg_L = L + 1 + \frac{(1-wq)^{L+1} - 1}{wq}$$

Na rysunku 6 pokazano zależność pomiędzy avg_L, czyli średnim rozmiarem kolejki a ilością pakietów *L* (oś y od 0 do 100 pakietów) i wq (*ang. queue weight*) kolejką zadań (oś x od 0,001 do 0,005). Przykładowo dla wartości *wq* = 0,001, po zwiększeniu pakietów *L* od 0 do 100 średnia wielkość pakietu avg_{100} wyniesie 4,88 pakietu.

6. Podsumowanie

Algorytm RED nie jest lekarstwem na wszystko, aplikacje, które niewłaściwie realizują wykładniczy rozkład, wciąż mają nieuczciwy udział w przepustowości, jednak dzięki RED nie powodują one tak dużej szkody dla przepustowości i opóźnień innych połączeń.
RED statystycznie upuszcza pakiety z przepływów, zanim osiągnie swój twardy limit. Powoduje to, że zatłoczone łącze szkieletu zwolni bardziej płynnie i zapobiega też synchronizacji retransmisji. Pomaga to również w szybszym odczytywaniu przez TCP „uczciwej" prędkości, pozwalając na odrzucenie niektórych pakietów wcześniej, utrzymując niskie rozmiary kolejek i kontrolując opóźnienia.
Prawdopodobieństwo, że pakiet zostanie usunięty z określonego połączenia, jest proporcjonalne do jego wykorzystania przepustowości, a nie do liczby pakietów, które przesyła.
RED to dobra opcja dla rozbudowy routerów w konfiguracji szkieletowej, w której nie można sobie pozwolić na złożoność śledzenia stanu w sesji potrzebnego dla odpowiedniego kolejkowania.
Algorytm znalazł zastosowanie przy przesyłaniu dużych pakietów danych podczas analizy termograficznej w inżynierii biomedycznej. Szczególnie w fizykoterapii gdzie termowizja wykorzystywana jest do oceny skuteczności wykonywanych zabiegów do śledzenia i oceny zabiegów seryjnych [12].
Czysty algorytm RED nie uwzględnia zróżnicowania jakości usług (*QoS ang. quality of service*).
Istnieją odmiany algorytmu poprawiające jego działanie [5, 11, 13, 14]:
Weighted random early detection (WRED) zapewnia wczesne wykrywanie z uwzględnieniem QoS, Robust random early detection (RRED) zapewnia zwiększenie przepustowości TCP przeciwko atakom Denial-of-Service (DoS), w szczególności atakom niskopoziomowym typu Denial-of-Service.

Literatura

[1] D. Agrawal, N.L.S. da Fonseca and F. Granelli, "Integrated ARM/AQM mechanisms based on PID controllers," Proc. ICC, pp. 6-10, May 2005.
[2] S. Athuraliya, S.H. Low, V.H. Li and Q. Yin, "REM: Active queue management," IEEE Network Mag., vol. 15, no. 3, pp. 48-53, 2001
[3] B. Braden, et al., "Recommendations on queue management and congestion avoidance in the Internet," IETF RFC2309, 1998
[4] G. Chen and T. T. Pham, "Introduction to fuzzy systems," Chapman & Hall/CRC, 2006.
[5] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith. "Tuning RED for web traffic," IEEE/ACM Trans. Networking, vol. 9, No. 3, pp. 249-264, 2001.
[6] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," Proc. INFOCOM, pp. 1320-1328, Mar. 1999.
[7] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," available at http://www.icir.org/floyd/red.html
[8] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Networking, vol. 1, no. 4, pp. 397-413, Aug. 1993.
[9] C. V. Hollot, V. Misra, D. Towsley and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," Proc. IEEE INFOCOM 2001, vol. 3, Anchorage, Alaska, pp. 1726-1734, April 2001.
[10] L. Hu and A. D. Kshemkalyani, "HRED: A simple and efficient active queue management algorithm," Proc. ICCCN 2004, pp. 387-393, 2004.
[11] C. Joo, S. Bahk and S. S. Lumetta, "Hybrid active queue management," Proc. ISCC'03, pp. 999-1004, 2003.
[12] P. Kardasz i inni Termografia w inżynierii biomedycznej Laboratorium-Przegląd Ogólnopolski, 31-38
[13] J. Koo, K. Chung, H. Kim and H. Lee, "A new active RED algorithm for congestion control in IP networks" LNCS 2343, pp. 469-479, 2002.
[14] S. S. Kunniyur and R. Srikant, "An adaptive virtual queue (AVQ) algorithm for active queue management," IEEE/ACM Transactions on Networking, vol. 12, no. 2, pp. 286-299, April 2004.