

STEROWANIE RUCHEM W SIECIACH TELEKOMUNIKACYJNYCH

*Zapobieganie przeciążeniom
w sieciach telekomunikacyjnych*

Przeciążenie

- **Stan długotrwałego przeładowania sieci, kiedy zapotrzebowanie na ograniczone jej zasoby jest zbliżone do przepustowości sieci lub ją przekracza**
- **Konsekwencje przeciążenia**
 - ✓ **upadek sieci (niezdolność do świadczenia usług na zadanym poziomie)**
 - ✓ **niedostarczanie pakietów, które są odrzucane przed ich dostarczeniem do węzła docelowego**
 - ✓ **transmitowanie fragmentów pakietów, które są odrzucane przed wykorzystaniem ich do odtworzenia pakietów**
 - ✓ **transmitowanie pakietów nieprzydatnych dla użytkownika (z powodu ich nadmiernego opóźnienia)**

Degradowane zasoby

✓ ***przepustowość kanałów transmisyjnych***

- ↳ *niewykorzystanie dostępnej przepustowości*
- ↳ *wykorzystanie przepustowości na potrzeby retransmisji pakietów*

✓ ***pojemność pamięci buforowych***

- ↳ *niewykorzystanie pojemności buforów*
- ↳ *zawłaszczenie pojemności buforów w oczekiwaniu na zakończenie obsługi (transmisji pakietów)*

Przyczyny degradacji zasobów

- ✓ *konflikt żądań dostępu do zasobów uniemożliwiającego skorzystanie z nich do chwili jego rozstrzygnięcia*
 - ↳ *kolizyjne reguły dostępu*
- ✓ *niedopasowanie zasobów*
 - ↳ *problemy wymiarowania sieci*
- ✓ *nieprawidłowe przyznanie zasobów w porównaniu z zapotrzebowaniem lub możliwością ich wykorzystania*
 - ↳ *zawłaszczanie zasobów powodowane np. agresywną strategią dostępu*

Zakleszczenia

- ***Zakleszczenie (ang. deadlock) – sytuacja, w której co najmniej dwie różne akcje czekają na siebie nawzajem, więc żadna nie może się zakończyć***
- ***Warunki wystąpienia zakleszczenia***
 - ↳ ***wzajemne wykluczenie - jest możliwy jednoczesny równoległy dostęp wielu zadań do zasobu, lecz liczba jednocześnie zadanych żądań do zasobu jest większa, o co najmniej jeden, od liczby maksymalnych równoległych dostępów do zasobu, które mogą zostać obsłużone***
 - ↳ ***trzymanie zasobu i oczekiwanie - zadanie utrzymuje jeden z zasobów, ale do ukończenia pracy niezbędne jest także zaalokowanie zasobów innego typu***
 - ↳ ***cykliczne oczekiwanie - zadania w taki sposób żądają zasobów, że powstaje cykl***
 - ↳ ***brak wywłaszczania z zasobu - zadania dobrowolnie nie rezygnują z przydzielonych im zasobów; zwolnienie zasobów możliwe jest po zakończeniu zadania***

Źródła zakleszczeń

- ✓ *ograniczona przepustowość kanału*
- ✓ *ograniczona pojemność pamięci buforów*
- ✓ *zakleszczenie kompetencji*
- ✓ *zakleszczenie sekwencji*

Zapobieganie przeciążeniom

- *Celem metod zapobiegania powstawaniu przeciążeń jest taka obsługa ruchu, aby:*
 - ✓ *zapobiegać wprowadzeniu do sieci ruchu nadmiarowego w porównaniu ze zdolnością obsługową sieci*
 - ✓ *efektywnie obsługiwać ruch wprowadzony do sieci*

Mechanizmy zapobiegania przeciążeniom

- *Mechanizmy prewencyjne*
 - ↳ *zapobieganie przeciążeniom w wyniku podejmowania działań uprzedzających*

- *Mechanizmy reakcyjne*
 - ↳ *odtworzenie stanu sieci poprzedzającego powstanie stanu przeciążenia*

Metody zapobiegania przeciążeniom

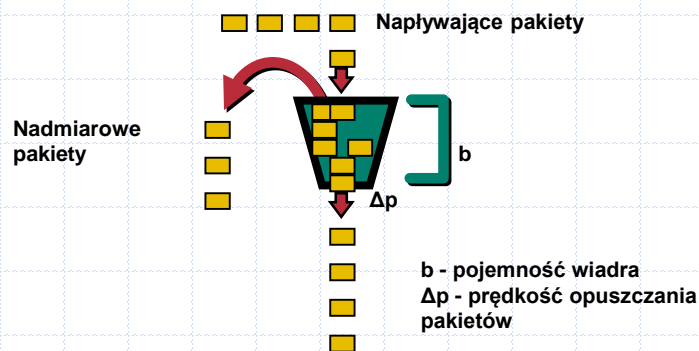
- *planowanie i udostępnianie zasobów*
- *sterowanie przyjmowaniem zgłoszeń*
- *kształtowanie ruchu*
- *alokacja szybkości transmisji*
- *selektywne odrzucanie*

Sterowanie przyjmowaniem zgłoszeń

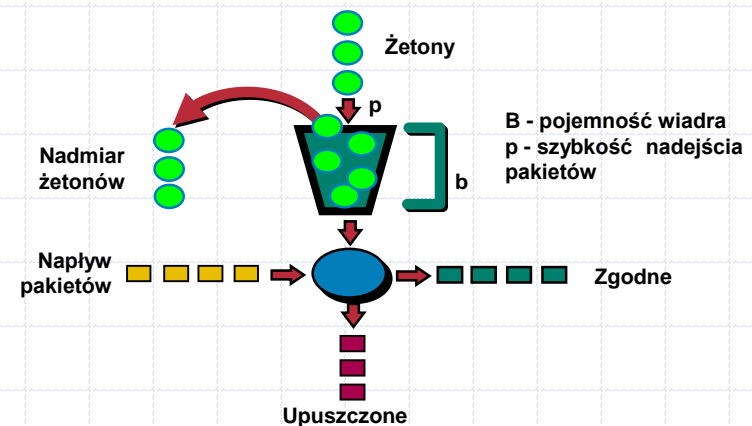
- *ograniczenie wielkości ruchu wejściowego (mechanizm sterowania przyjmowaniem zgłoszeń)*
 - ↪ *dostosowanie rodzaju i parametrów ruchu wprowadzanego do sieci poprzez podejmowanie decyzji o akceptacji lub odrzuceniu strumienia ruchu*

Kształtowanie ruchu

- stosowanie regulatora szybkości wprowadzania jednostek danych do sieci (mechanizm wygładzania ruchu)
 - ➔ szybkość wprowadzania jednostek danych, nie większa od szybkości generowania danych w źródle, jest wielkością podlegającą sterowaniu

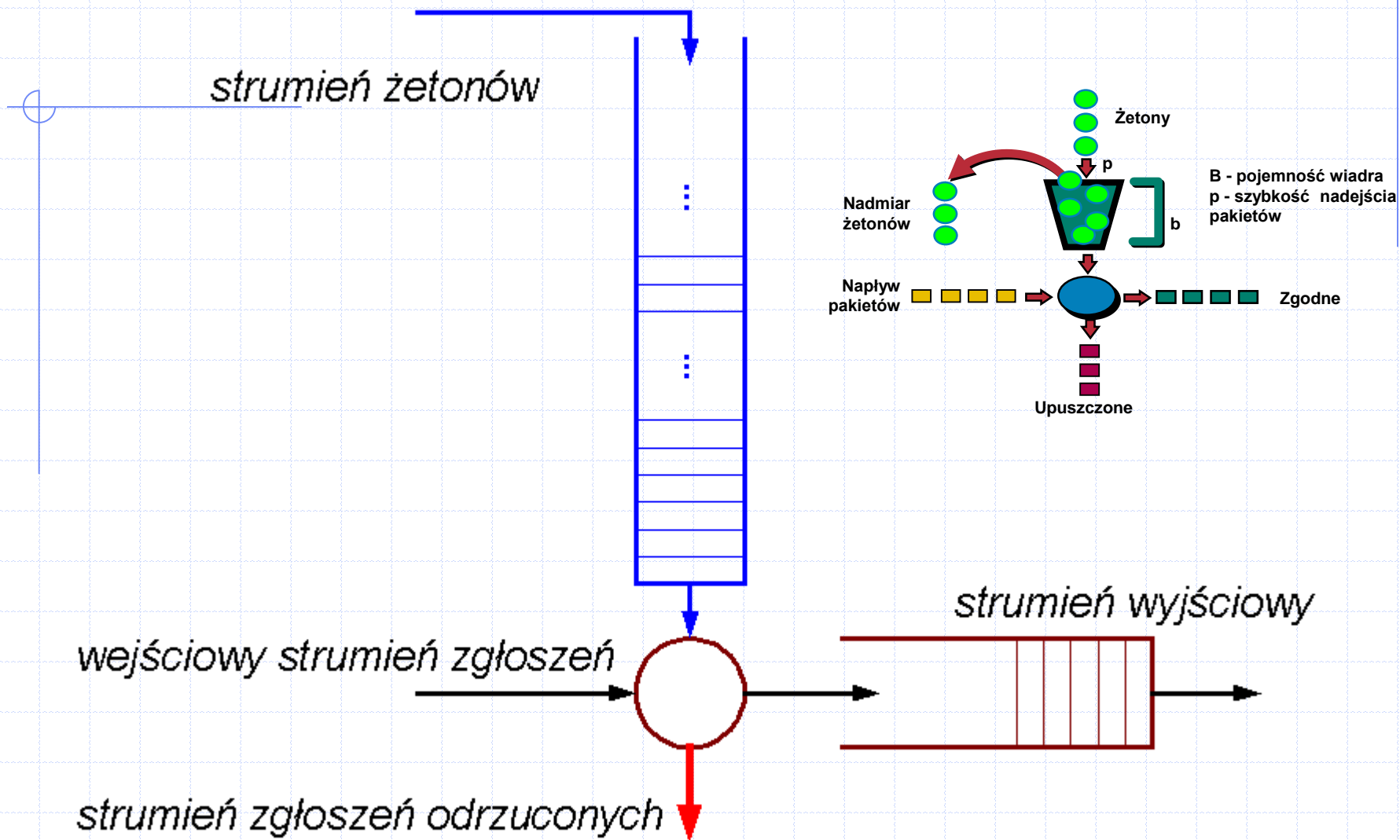


Model ciekącego wiadra

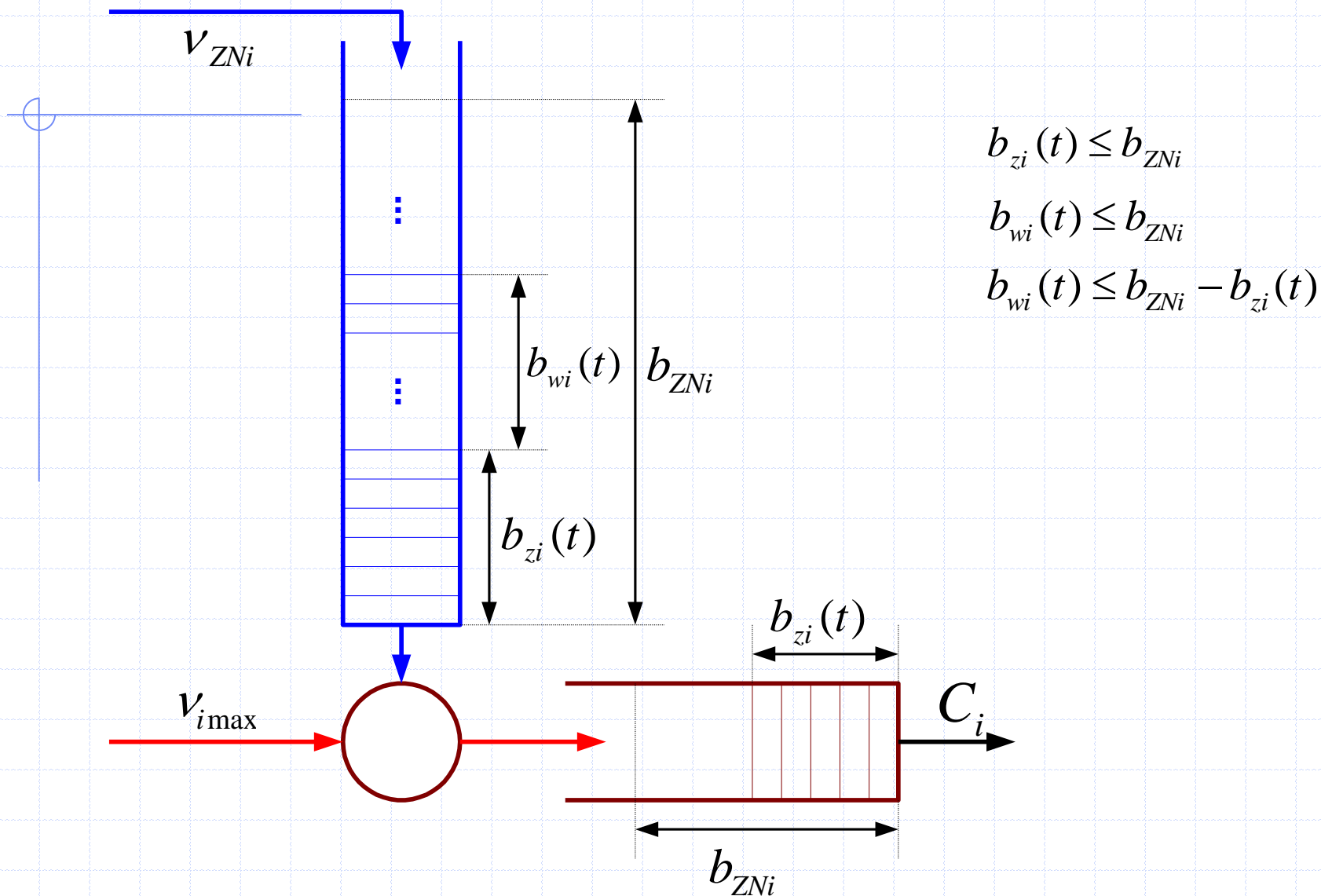


. Model wiadra z żetonami

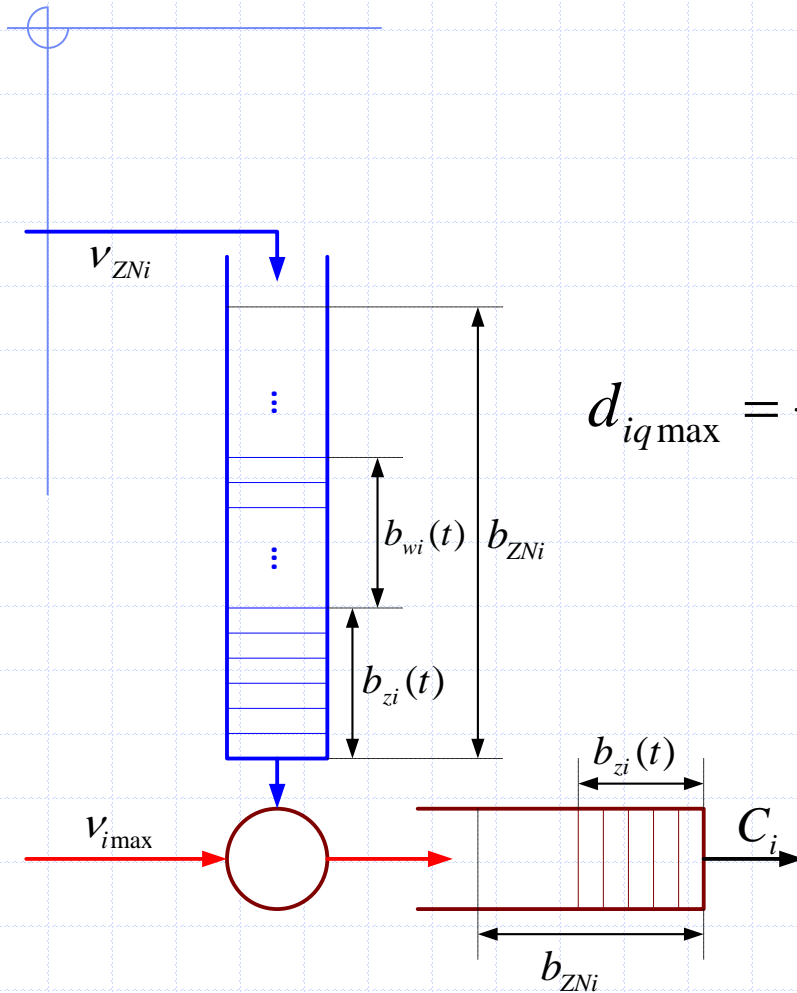
Mechanizm Token Bucket (wiadro żetonów)



Sterowanie opóźnieniem



Sterowanie opóźnieniem



$$d_{i\max} = d_{iq\max} + d_{ipac} + d_{ipro}$$

$$d_{iq\max} = \begin{cases} \frac{b_{ZNi}}{C_i} & \text{gdy } v_{i\max} \rightarrow \infty, \quad C_i \geq v_{ZNi} \\ \frac{b_{ZNi}(v_{i\max} - C_i)}{C_i(v_{i\max} - v_{ZNi})} & \text{gdy } v_{i\max} > C_i \geq v_{ZNi} \\ 0 & \text{gdy } v_{ZNi} \leq v_{i\max} \leq C_i \\ \frac{b_{ZNi}}{C_i} & \text{gdy } v_{ZNi} > C_i \end{cases}$$

Przeciwdziałanie przeciążeniom w sieciach TCP/IP

Parametry kredytowego mechanizmu okna

- ✓ *rozmiar okna*
- ✓ *długość zwłoki czasowej retransmisji*

Procedury stosowane w mechanizmie okna

- ✓ *procedura spowolnionego startu*
- ✓ *procedura dynamicznego wymiarowania okna*
- ✓ *procedura szybkiej retransmisji*
- ✓ *procedura szybkiego odtwarzania przepustowości*

Alokacja szybkości transmisji w połączeniu TCP

Funkcja celu:

$$Q(v) = g(v) - pv$$

koszt uzyskania pojemności

gdzie:

$$g(v) = k - \frac{2}{rtt^2 v}$$

cena przeciążenia

$$v = \frac{1}{rtt} \sqrt{\frac{2}{p}}$$

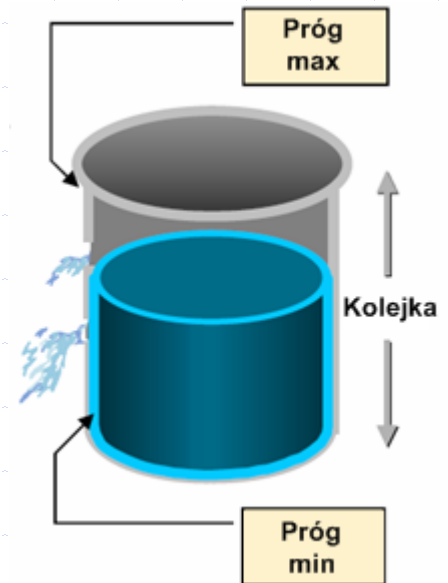
szybkość transmisji segmentów
w pojedynczym połączeniu

rtt – wartość opóźnienia potwierdzenia segmentu (round-trip time)

p – prawdopodobieństwo straty segmentu

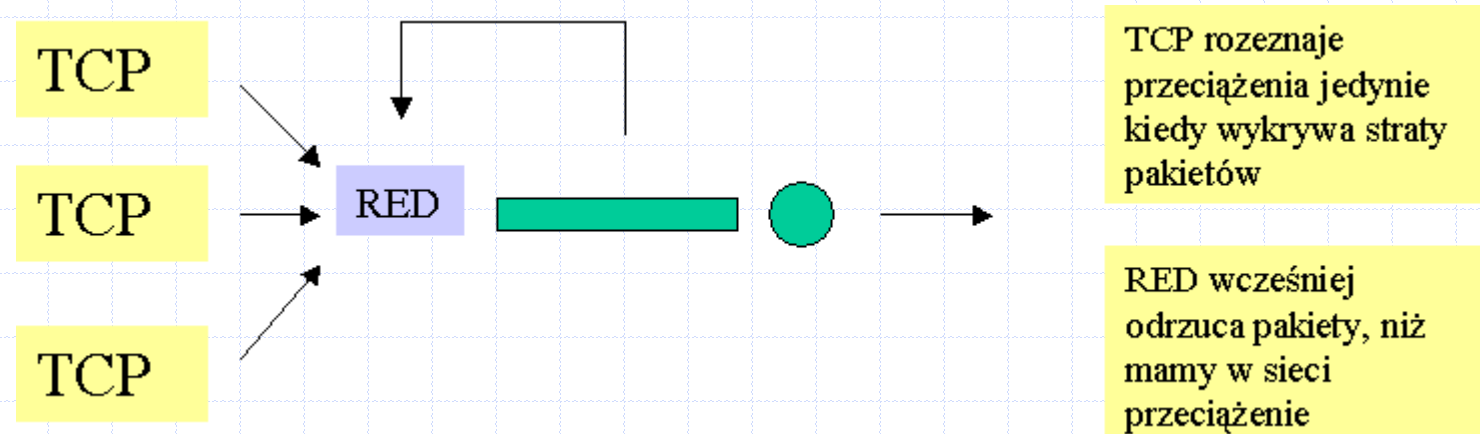
Losowe wczesne wykrywanie (Random Early Detection - RED)

- ✓ *mechanizm unikania przeciążeń w sieci z komutacją pakietów*
- ✓ *mechanizm wykrywa możliwość wystąpienia przeciążenia w sieci poprzez obliczanie średniej długości kolejki pakietów w buforze*
- ✓ *kiedy średnia długość kolejki przewyższa zadany próg, wówczas mechanizm odrzuca lub oznacza przybywające pakiety z pewnym prawdopodobieństwem, przy czym wartość tego prawdopodobieństwa jest funkcją średniej długości kolejki*



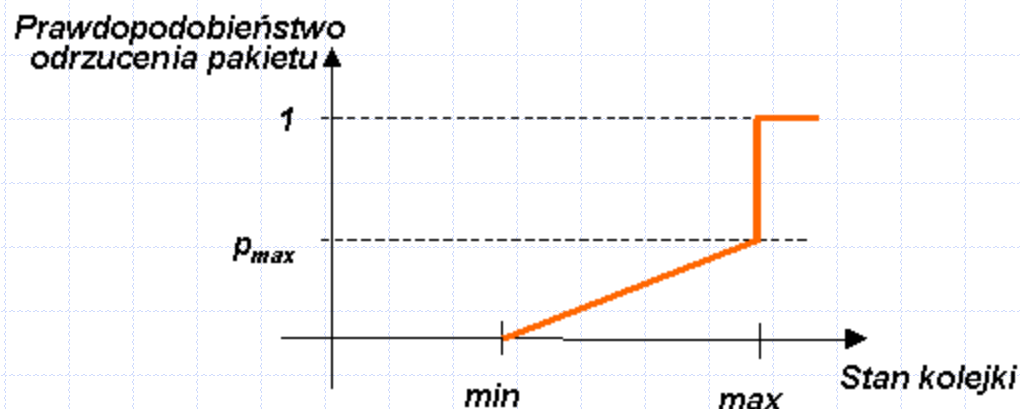
Mechanizm RED

- ✓ mechanizm RED utrzymuje średnią długość kolejki na niskim poziomie, ale pozwala sporadycznie na przyjmowanie przez kolejkę paczek pakietów
- ✓ w czasie przeciążenia, prawdopodobieństwo odrzucania pakietów w ramach poszczególnych połączeń jest „mniej więcej” sprawiedliwe, tzn. jest wprost proporcjonalne do tego w jaki sposób połączenia dzielą pomiędzy sobą zasoby
- ✓ RED został zaprojektowany do współpracy z mechanizmami sterowania przed przeciążeniami w warstwie transportowej (TCP)
- ✓ w efekcie końcowym, RED zapobiega negatywnemu efektowi „synchronizacji” źródeł TCP (czyli jednoczesnemu zmniejszaniu rozmiaru okna w źródłach TCP w przypadku występowania przeciążenia)



Algorytm RED

- ✓ algorytm RED oblicza średnią długość kolejki przy użyciu tzw. metody filtru dolnoprzepustowego z wykładniczym ważonym uśrednianiem przesuwającym się
- ✓ średnia długość kolejki jest porównywana z dwoma wartościami progowymi, progiem minimalnym i progiem maksymalnym
- ✓ gdy średnia długość kolejki jest mniejsza od progu minimalnego, wówczas pakiety nie są tracone
- ✓ gdy średnia długość kolejki jest większa niż wartość maksymalna, wówczas każdy pakiet jest markowany; jeśli markowane pakiety są odrzucane, wówczas średnia długość kolejki nie przewyższa maksymalnego progu
- ✓ jeśli średnia długość kolejki jest pomiędzy wartością minimalną i maksymalną pakiety są markowane z pewnym prawdopodobieństwem



Algorytm RED

Dla każdego przychodzącego pakietu:

Oblicz średnią długość kolejki l_{avg}

If $l_{min} \leq l_{avg} < l_{max}$

Oblicz prawdopodobieństwo $p_a(l_{avg})$

Oznacz pakiet z prawdopodobieństwem $p_a(l_{avg})$

else if $l_{max} \leq l_{avg}$

Oznacz pakiet

Mamy zatem dwa niezależne algorytmy:

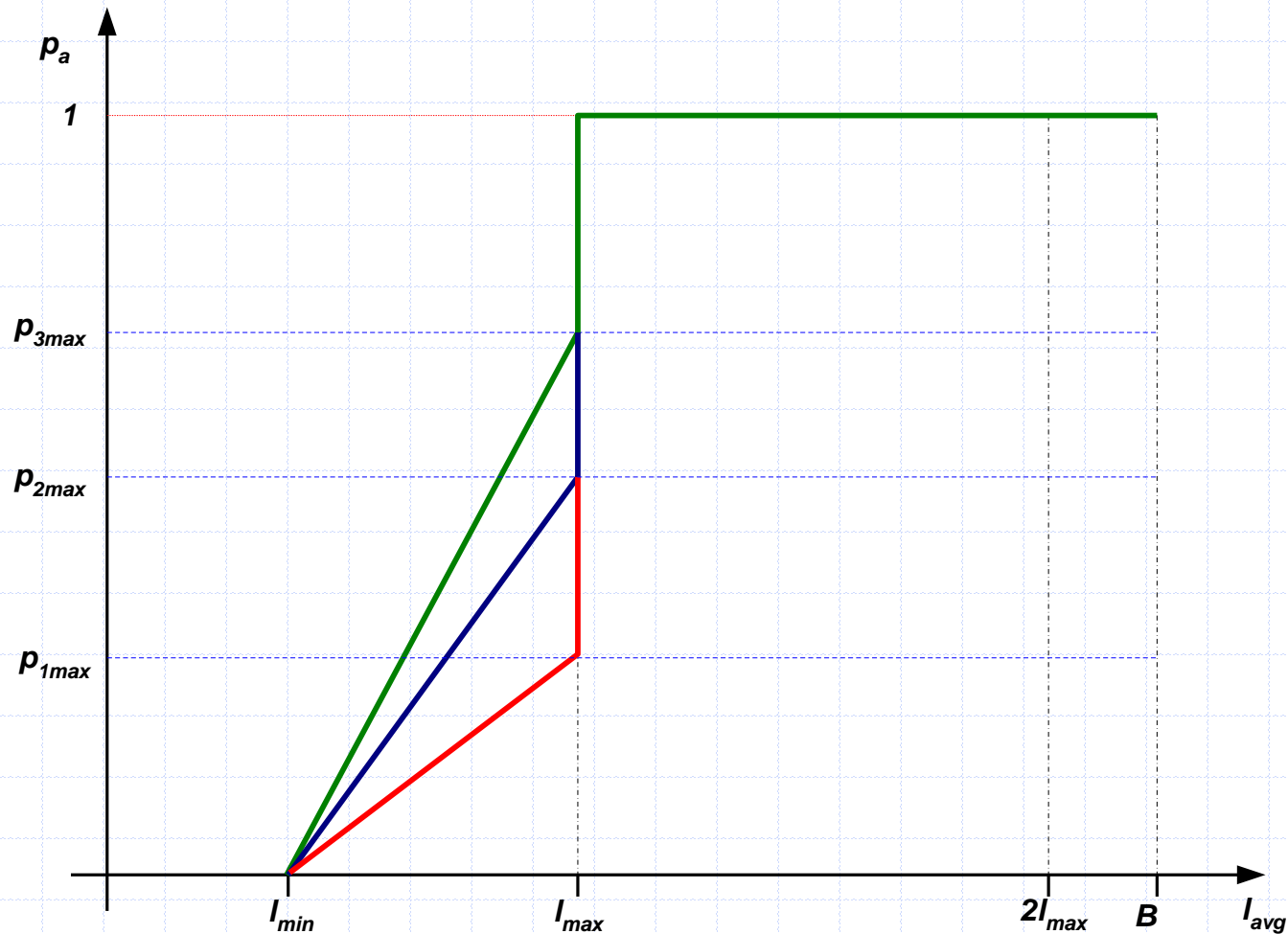
- (1) wyznaczenia średniej długości kolejki*
- (2) wyznaczenia prawdopodobieństwa znakowania*

Algorytm RED – funkcja znakowania

$$p_a(l_{avg}) = \begin{cases} 0 & gdy & 0 \leq l_{avg} < l_{min} \\ \frac{l_{avg} - l_{min}}{l_{max} - l_{min}} p_{max} & gdy & l_{min} \leq l_{avg} < l_{max} \\ 1 & gdy & l_{max} \leq l_{avg} \leq B \end{cases}$$

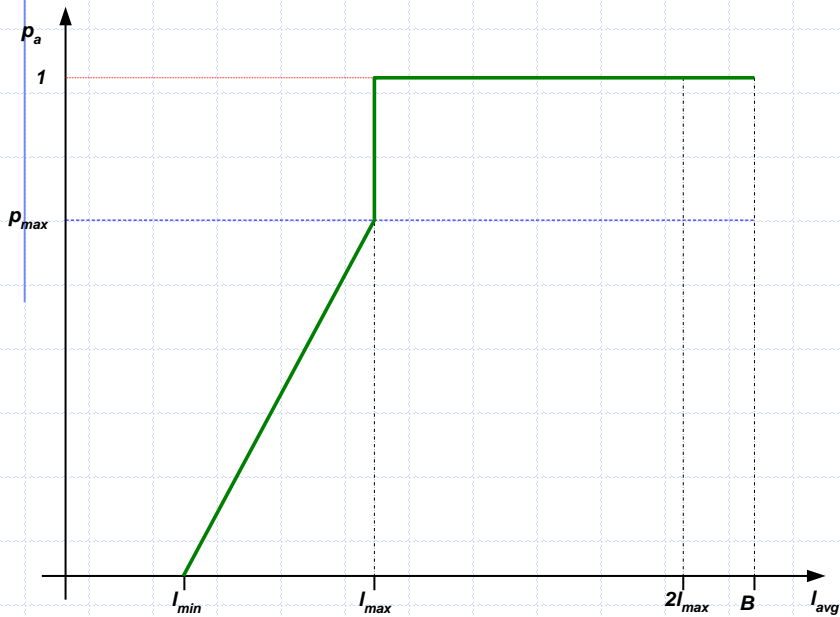
Algorytm WRED (Weighted Early Detection)

- Mechanizm RED stosowany oddzielnie dla każdego strumienia

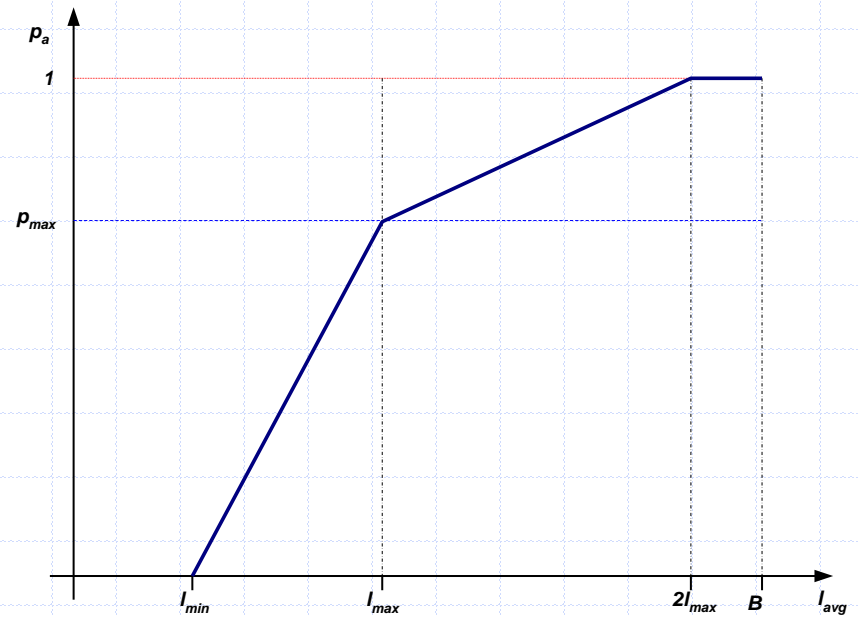


Algorytm G-RED (Gentle RED)

➤ Mechanizm łagodnego odrzucania (znakowania) pakietów



RED



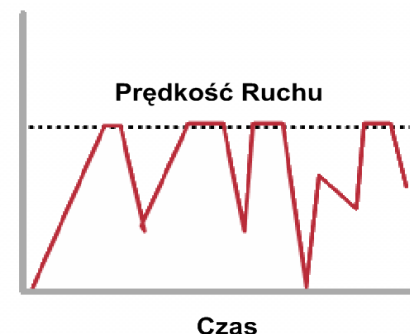
G-RED

Algorytm G-RED – funkcja znakowania

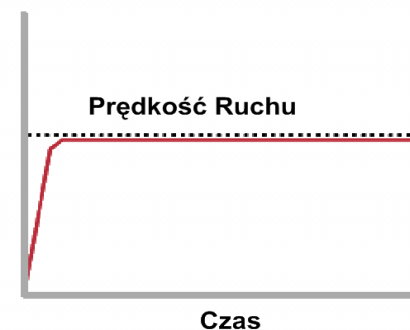
$$p_a(l_{avg}) = \begin{cases} 0 & \text{gdy } 0 \leq l_{avg} < l_{min} \\ \frac{l_{avg} - l_{min}}{l_{min} - l_{max}} p_{max} & \text{gdy } l_{min} \leq l_{avg} < l_{max} \\ p_{max} + \frac{l_{avg} - l_{max}}{l_{max}} (1 - p_{max}) & \text{gdy } l_{max} \leq l_{avg} < 2l_{max} \\ 1 & \text{gdy } l_{max} \leq l_{avg} \leq B \end{cases}$$



Odrzucanie



Wygladzenie

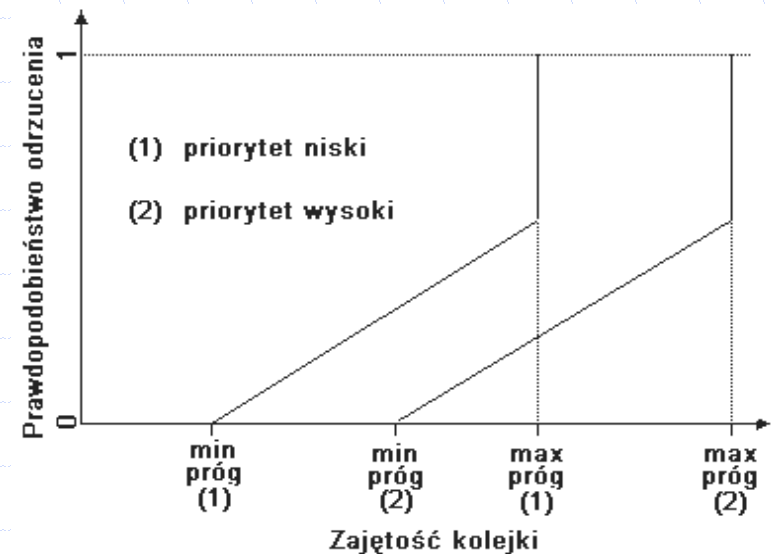
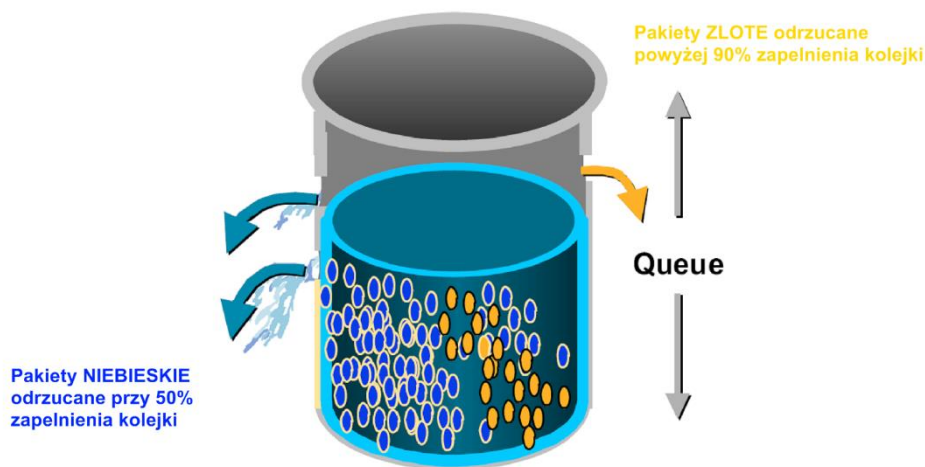


Algorytm WRED

Najbardziej znaną odmianą mechanizmu RED jest algorytm „ważonego losowego wczesnego wykrywania” (ang. Weighted Random Early Detection).

Różni się on od algorytmu RED zdolnością rozpoznawania różnych klas ruchu, dla których indywidualnie definiuje progi wypełnienia kolejki oraz maksymalne prawdopodobieństwo odrzucenia pakietu.

Przez to mechanizm ten może zapobiegać przeciążeniom sieci, odrzucając w pierwszej kolejności ruch mniej istotny.





Dziękuję za uwagę!