

WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

Modyfikacja algorytmów retransmisji protokołu TCP.

Student

Adam Markowski

Promotor

dr hab. Michał Grabowski

Cel pracy

Celem pracy było przetestowanie i sprawdzenie przydatności modyfikacji klasycznego algorytmu obliczania czasu oczekiwania na potwierdzenie (RTO) w porównaniu z algorytmem klasycznym.

Przewidziano wykonanie eksperymentu wstępnego oraz eksperymentu właściwego. Następnie porównując uzyskane wyniki dla algorytmu klasycznego oraz zmodyfikowanego udało się wyciągnąć serię wniosków, które pozwoliły ocenić przydatność zaimplementowanej w środowisku testowym modyfikacji.

Kolejnym celem pracy było przybliżenie czytelnikowi problemu optymalizacji algorytmów adaptacyjnych oraz pokazanie jak ważne jest to zagdanie oraz jak rozwijało się na przestrzeni lat.



Algorytmy klasyczne obliczania czasu oczekiwania na potwierdzenie

Protokół TCP został tak skonstruowany, aby dawał pewność połączenia między nadawcą a odbiorcą. Do tego celu wykorzystywany jest mechanizm potwierdzenia z retransmisją. Działa on w ten sposób, że wysyłane przez oprogramowanie warstwy aplikacji bajty są kolejno numerowane. Następnie protokół TCP rozdziela te dane na fragmenty i umieszcza je w segmentach. Segmenty są oznaczane kolejnymi numerami sekwencyjnymi, po to aby odbiorca mógł zestawzić odebrane dane w odpowiedniej kolejności. W kolejnym kroku tak przygotowane segmenty są umieszczane w kolejce nadawczej i wysyłane do odbiorcy. W tym momencie także ustawiany jest zegar RTO (Retransmission time out) odliczający czas w dół (do zera). O ile w tym czasie nie dotrze potwierdzenie prawidłowego odebrania segmentu od odbiorcy następuje ponowne wysłanie niepotwierdzonego segmentu do odbiorcy.

Czas oczekiwania na potwierdzenie jest zmienny, ponieważ zależy od bieżącego stanu sieci i wyliczany jest w ten sposób, że TCP cały czas sprawdza czas podróży pakietów i z tych właśnie próbek wylicza aktualny czas oczekiwania na potwierdzenie (ang. RTO – Retransmission Time Out).

Najstarsza wersja protokołu TCP miała zaimplementowany algorytm wyliczania czasu oczekiwania na potwierdzenie oparty o pojedyncze wygładzanie wykładnicze, natomiast kolejna wersja specyfikacji TCP, także uznawana w tej chwili za klasyczną miała już zaimplementowane podwójne wygładzanie wykładnicze i zastosowanie tego nowego algorytmu przyczyniło się do znacznego wzrostu przepływności w sieci.



Algorytm klasyczny obliczania czasu oczekiwania na potwierdzenie

Algorytm klasyczny:

Różnica = próbka_czasu - stare_RTT

NOWE RTT = stare_RTT + alfa*Różnica (= (1 - alfa)*stare_RTT + alfa*próbka_czasu)

Odchylenie = stare_Odchylenie + beta*(| Różnica | - stara_Różnica)

Czas_oczekiwania RTO = RTT + eta * Odchylenie



Współczynniki :

alfa - współczynnik wygładzania sterujący tym jak silny jest wpływ wartości nowej próbki czasowej na prognozę RTT oraz jak wpływa na poprzednio obliczone RTT.

beta - wartość tego współczynnika ma wpływ na obliczane odchylenie błędu z ostatnich dwóch prognoz RTT.

eta - od wartości tego współczynnika zależy wpływ prognozy odchylenia na nowo obliczaną wartość RTO.

Heurystyka :

1. Jeżeli poprzednie odchylenie próbki jest w dół (stara_Różnica < 0) to przyjmowane odchylenie jest zwiększane (ustalane przez współczynnik beta).
2. Jeżeli poprzednie odchylenie próbki jest w górę (stara_Różnica > 0), wartość większa od wartości bezwzględnej aktualnego odchylenia próbki, to przyjmowane odchylenie jest zmniejszane (ustalane przez współczynnik beta).
3. Jeżeli poprzednie odchylenie jest w górę, ale wartość mniejsza od wartości bezwzględnej aktualnego odchylenia próbki, to przyjmowane odchylenie jest zwiększane (ustalane przez współczynnik beta).

Algorytm zmodyfikowany obliczania czasu oczekiwania na potwierdzenie

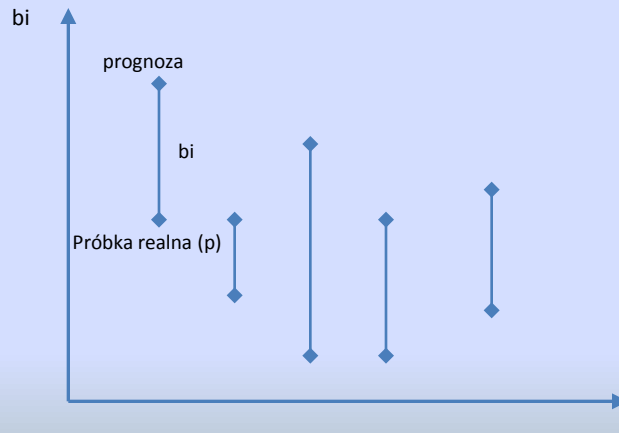
Algorytm zmodyfikowany:

W tym przypadku posłużymy się obliczaną dynamicznie wartością współczynnika alfa a będzie ona obliczana na podstawie n ostatnich wartości pobranych z szeregu czasowego.

Rozmiar n będzie wartością wejściową algorytmu. Zamienimy zatem dobieranie wartości wejściowej alfa o skali ciągłej na dobieranie wartości wejściowej n o skali dyskretnej.

Heurystyka:

Im błąd prognozy jest bardziej systematyczny dla szeregu ostatnich wartości n, tym większe oddziaływanie na kolejną liczoną prognozę ma ostatnia przed punktem czasowym następnej obliczanej prognozy, wartość szeregu czasowego.



$$1) \quad b_i = |p - \text{poprzednia prognoza}|$$

$$2) \quad S = \sum_{i=1}^5 b_i$$

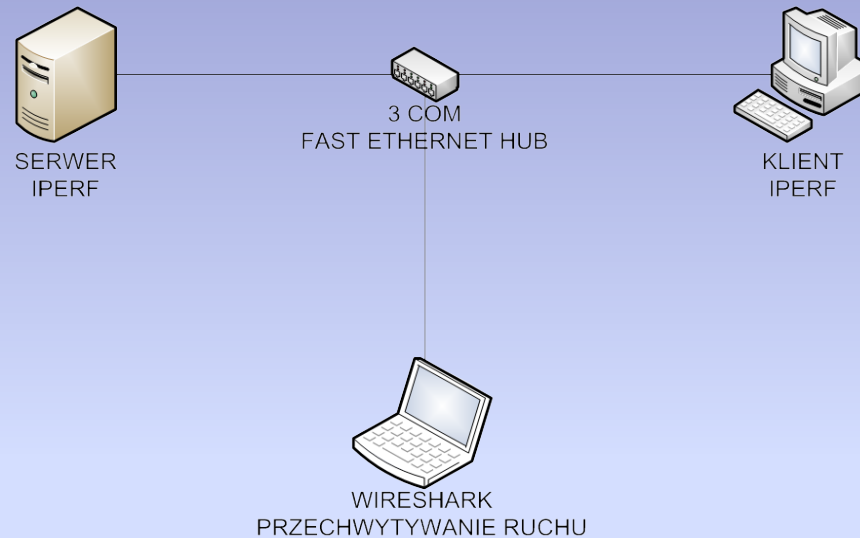
$$3) \quad \bar{b}_i = b_i / S$$

$$4) \quad \text{Alfa} = - \sum_{i=1}^5 \bar{b}_i * \log(\bar{b}_i)$$

p(wartość realna)	poprzednia prognoza b	b+	Alfa	
6,0000	8.784785516	2.7848	0.19758	0.13915
5.8310	8.784785516	2.9538	0.20958	0.14223
5.9161	8.784785516	2.8687	0.20354	0.14072
6,0000	8.784785516	2.7848	0.19758	0.13915
6.0828	8.784785516	2.702	0.19171	0.13752
		14.094		0.69877

Prognoza: 9.500452

Laboratorium eksperymentu – przechwytywanie ruchu w sieci



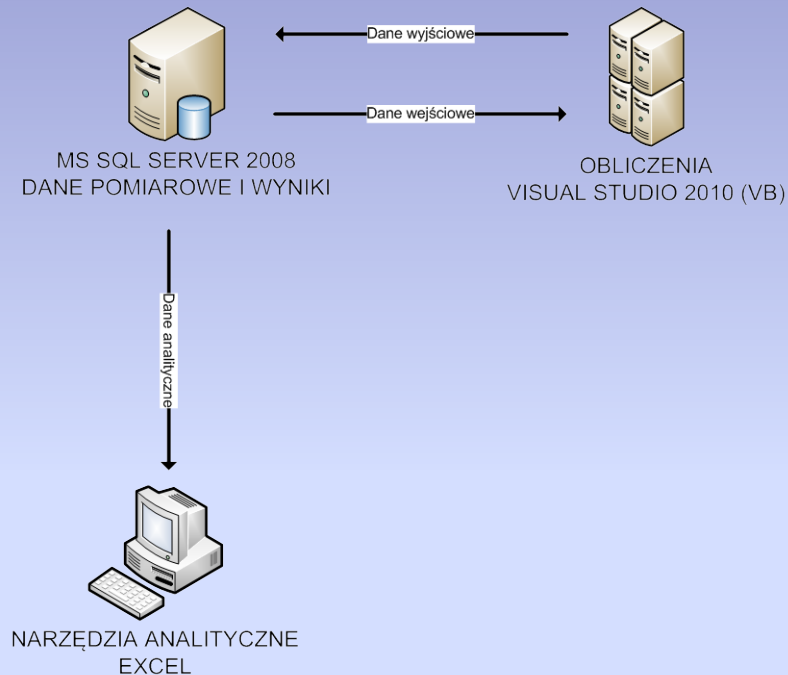
Laboratorium składa się z :

- Trzech komputerów z systemem operacyjnym Windows XP
- Hub Fast Ethernet firmy 3Com

Oprogramowanie :

- Wireshark
- Iperf (jako klient i serwer)
- Windump

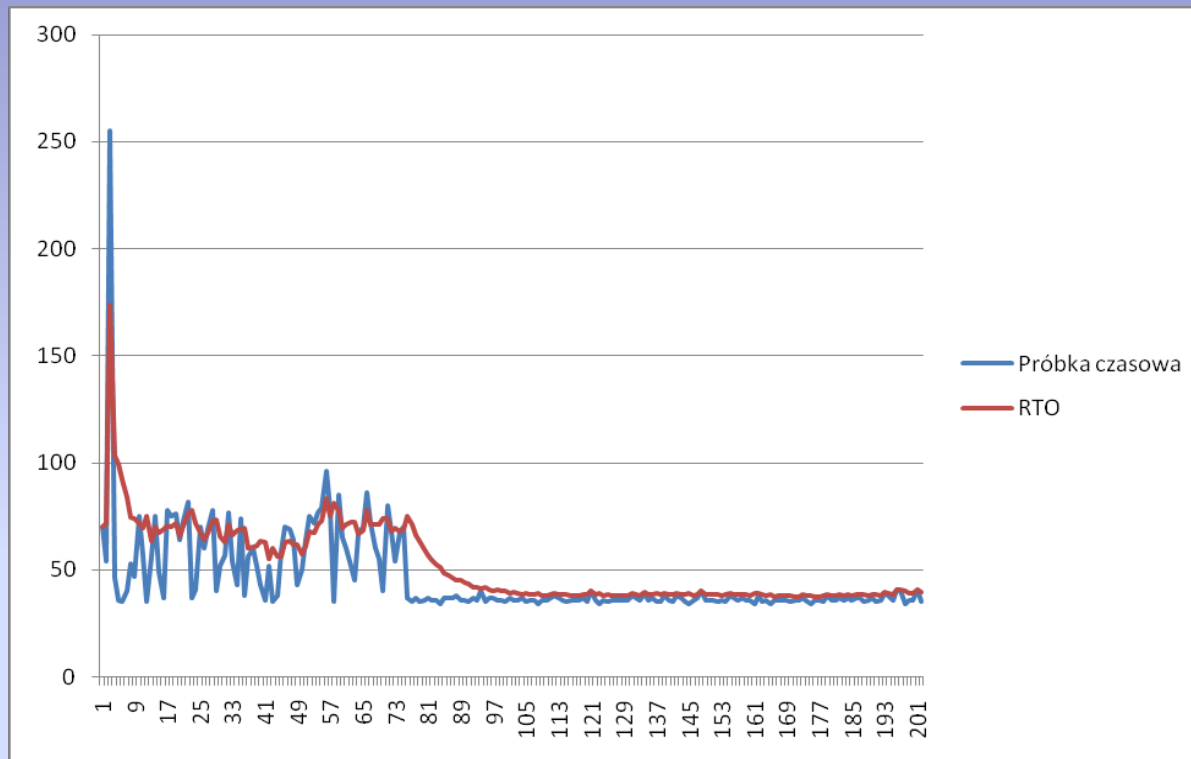
Laboratorium eksperymentu – część analityczna.



Część analityczna składa się z :

- Serwer baz danych MS SQL 2008
- Zasadniczy program liczący napisany w środowisku MS Visual Studio 2008 (wariant 1), algorytm zrealizowany w Excelu przy pomocy formuł (wariant 2)
- Nadzędzie analityczne (Excel 2007)

Wyniki eksperymentu



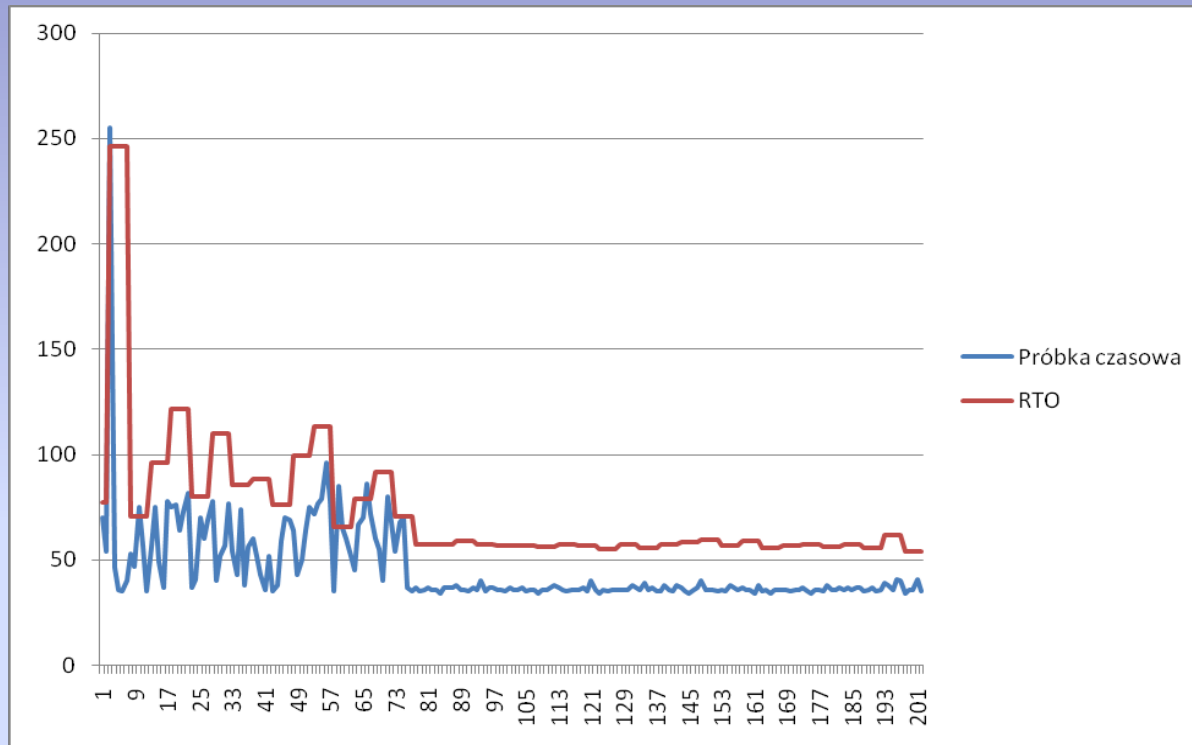
Algorytm klasyczny.

Wartości stałe : $\alpha=0,125$; $\beta=0,5$; $\eta=4$ (za książką [2] strona 41)

Wyniki testu pokazują bardzo dobrą adaptację algorytmu do zmian zachodzących w sieci (zmiany wartości próbek czasowych) co rozumiemy jako szybką reakcję na te zmiany. Algorytm klasyczny cechuje się przede wszystkim bardzo dobrze zbalansowaną zależnością pomiędzy ilością pojawiających się retransmisji oraz czasem propagacji przesyłanych przez sieć danych, co skutkuje także bardzo dobrą wydajnością.

Do cech ujemnych algorytmu klasycznego możemy zaliczyć niższą stabilność połączenia.

Wyniki eksperymentu

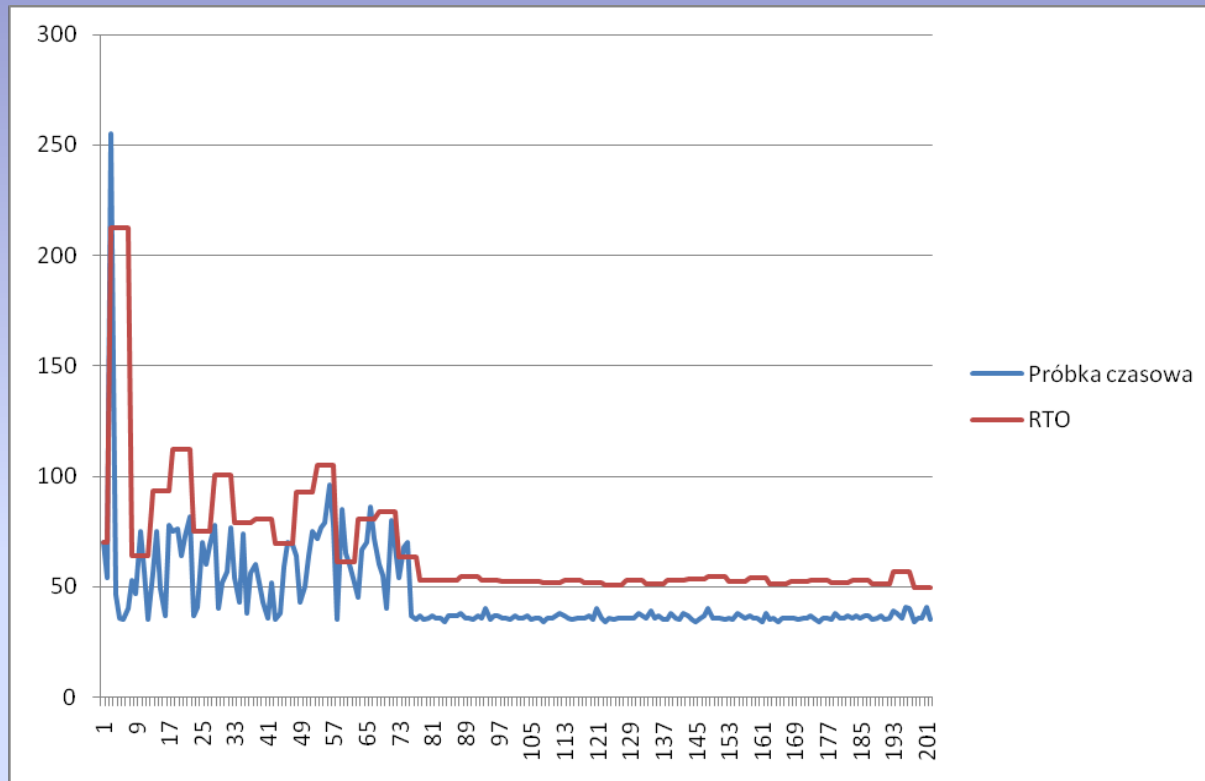


Algorytm zmodyfikowany – eksperyment 1.

Test przeprowadzony przy wartości alfa wyliczanej oraz $\beta = 0,125$

Porównując zachowanie algorytmu zmodyfikowanego z algorytmem klasycznym zauważamy zwiększony o 1,4 x czas propagacji ale także zmniejszoną 4-o krotnie ilość retransmisji spowodowanych osiągnięciem zera przez zegar retransmisji co skutkuje także wyższym poziomem stabilności połączenia.

Algorytm zmodyfikowany

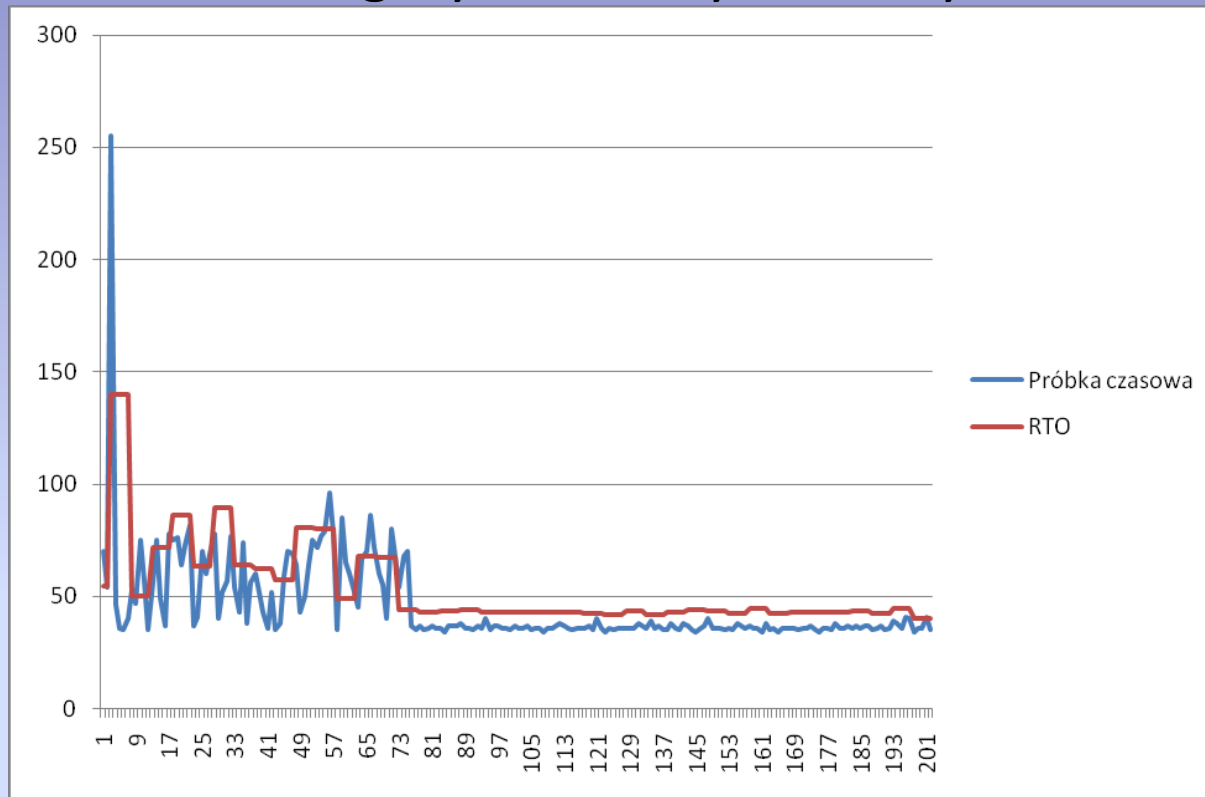


Algorytmy zmodyfikowany – eksperyment 2.

Test przeprowadzony przy wartości alfa wyliczanej oraz $\beta = 0,25$

W tym przypadku czas propagacji w stosunku do algorytmu klasycznego był o 1,28 x dłuższy a ilość retransmisji mniejsza o 2,95 x .

Algorytm zmodyfikowany



Algorytm zmodyfikowany – eksperyment 2.

Test przeprowadzony przy wartości alfa wyliczanej oraz $\beta = 0,51$

Ostatni z testów jest osiągniętymi wynikami najbardziej zbliżony do algorytmu klasycznego i charakteryzuje się nieznacznie większą ilością retransmisji, nieznacznie dłuższym czasem propagacji wysyłanych danych oraz trochę lepszą stabilnością połączenia w stosunku do algorytmu klasycznego

Zestawienie wyników

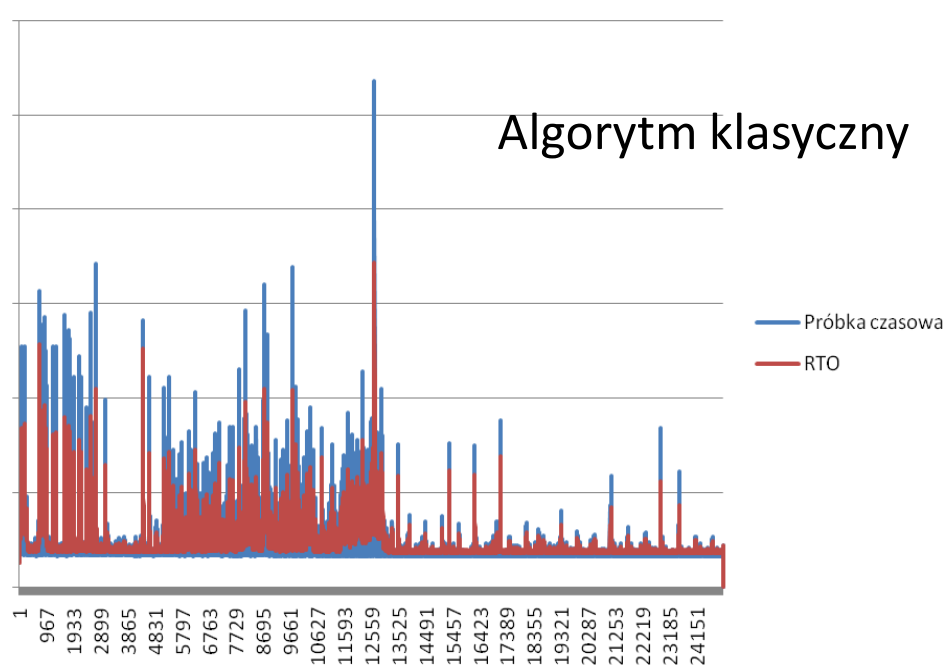
Nr testu	Algorytm	Wartość beta	Ilość retransmisji	Czas propagacji	Stabilność połączenia	BER
1	Klasyczny	0,125	2844	44,704	1,15	0,113
1A	Eksperyment 1	0,125	706	62,654	1,630	0,028
2	Eksperyment 2	0,25	963	57,394	1,494	0,038
3	Eksperyment 3	0,51	2892	45,383	1,184	0,115

Gdzie :

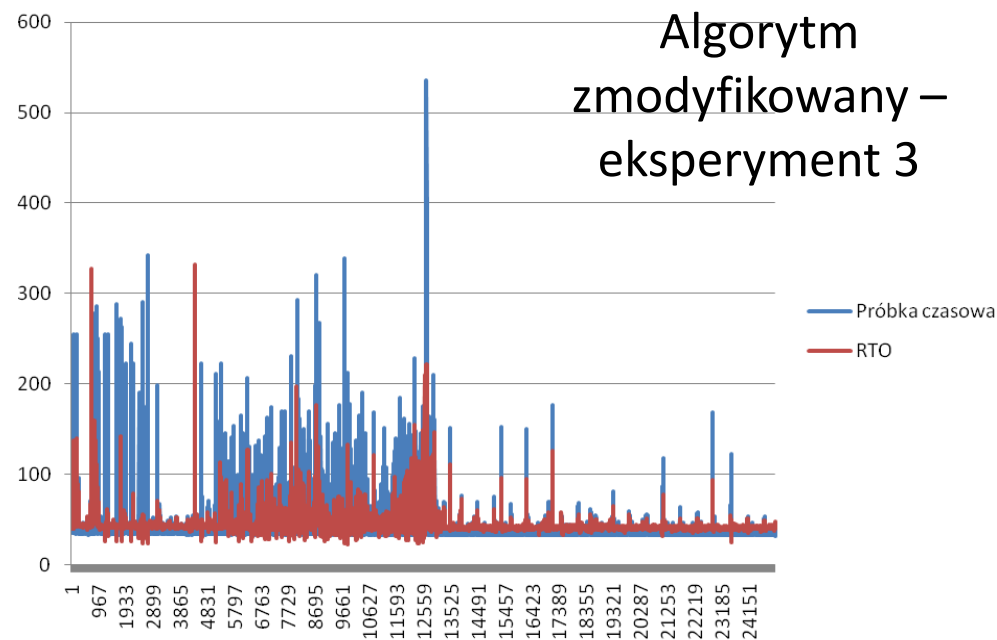
- **Ilość retransmisji** – ilość retransmisji spowodowanych przekroczeniem czasu oczekiwania na potwierdzenie (RTO).
- **Czas propagacji** – średni czas z sumy czasu oczekiwania na potwierdzenie z całej serii próbek.
- **Stabilność połączenia** – średnia wartość stosunku czasu oczekiwania na potwierdzenie (RTO) do próbki czasowej dla całego zbioru danych.
- **BER (z ang. Bit terror ratio)** – stosunek ilości bitów przesłanych z błędami do ogólnej liczby przesłanych bitów w określonym przedziale czasowym.

Porównanie zachowania algorytmów
dla całości próbek czasowych.

Algorytm klasyczny



Algorytm zmodyfikowany – eksperyment 3



Wnioski

- Wyniki uzyskane w toku eksperymentu są bardzo zbliżone do rezultatów jakie otrzymano przy wykorzystaniu algorytmu klasycznego a zależność między wynikami jest praktycznie liniowa.
- Minusem algorytmu zmodyfikowanego jest jego większa złożoność obliczeniowa co oznacza, że jego implementacja na sprzęcie fizycznym wymagała by zastosowania szybszych procesorów oraz w szczególności większej pamięci operacyjnej (np, w routerach).
- Według wstępnych obserwacji można przyjąć, że okno entropii powinno zmieniać swój rozmiar w zależności od tego jak stabilnie pracuje sieć. Najprawdopodobniej jego rozmiar powinien być znacznie większy w chwilach stabilnej pracy sieci a zmniejszać się radykalnie (nawet do rozmiaru $n=1$) w momentach, kiedy pojawiają się zakłócenia, zwiększa się wartość próbek czasowych i zaczynają pojawiać się retransmisje.
- Dodatkowym elementem, który w przyszłości mógłby wpływać na zachowanie się oraz wydajność algorytmu zmodyfikowanego, jest parametr beta. W toku eksperymentu, będącego treścią tej pracy został on przyjęty jako wartość stała. Trzy kolejne eksperymenty pokazały, że w sytuacji gdy zaczynają się zwiększać wartości próbek czasowych, wartość parametru beta również powinna rosnąć. Takie płynne zwiększanie wartości parametru beta owocowało by w tym przypadku zmniejszeniem się ilości retransmisji spowodowanych błędnie, to znaczy zbyt nisko, ocenionym czasem oczekiwania na potwierdzenie. Natomiast po wyrównaniu się parametrów pracy sieci i osiągnięciu przez nią optymalnych wartości próbek czasowych (oscylujących wokół średniej czasów z przyjętego interwału czasowego) radykalnie poprawił by się wtedy sposób wyliczania czasu oczekiwania na potwierdzenie, co jak wiadomo ma bezpośredni wpływ na ogólną wydajność sieci oraz stabilność jej pracy.

