

## RAPORT SYMULACJA CYFROWA

Mateusz Kwiatkowski  
M1 – przeglądanie działań  
Treść zadania

131092  
Protokół CSMA

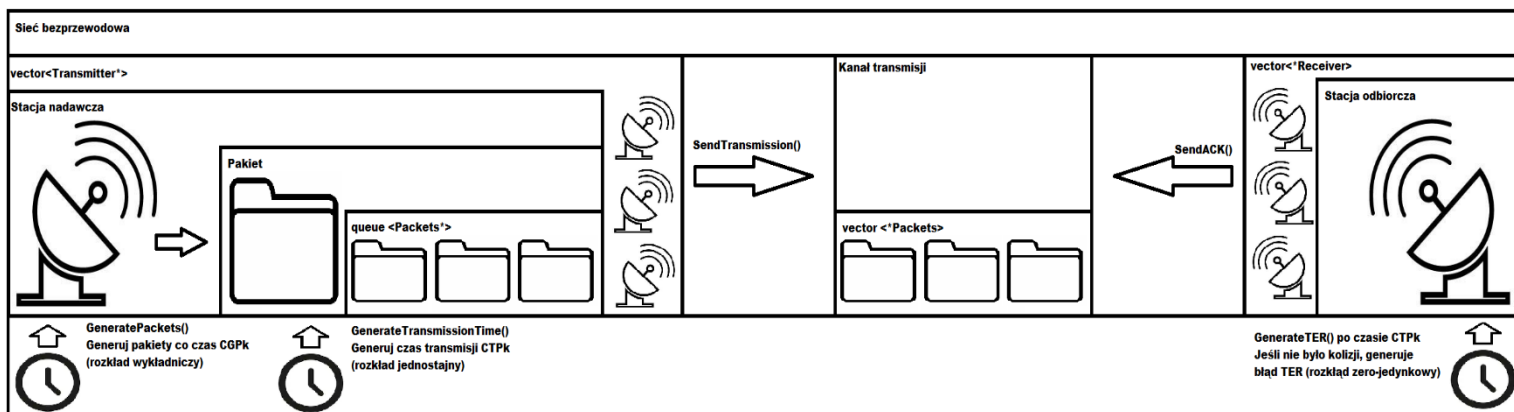
W sieci bezprzewodowej stacje nadawcze konkurują o dostęp do łącza. W losowych odstępach czasu CGPk k-ta stacja nadawcza generuje pakiety gotowe do wysłania. Po uzyskaniu dostępu do łącza zgodnie z algorytmem A, k-ty terminal podejmuje próbę transmisji najstarszego pakietu ze swojego bufora. Czas transmisji wiadomości z k-tej stacji nadawczej do k-tej stacji odbiorczej wynosi CTPk. Jeśli transmisja pakietu zakończyła się sukcesem, stacja odbiorcza przesyła potwierdzenie ACK (ang. Acknowledgment) poprawnego odebrania wiadomości. Czas transmisji ACK wynosi CTIZ. Jeśli transmisja pakietu nie powiodła się, stacja odbiorcza nie przesyła ACK. Odbiór pakietu uznajemy za niepoprawny, jeśli w kanale transmisyjnym wystąpiła kolizja lub błąd. Przez kolizję rozumiemy nałożenie się jakiegokolwiek części jednego pakietu na inny pakiet (pochodzący z innego nadajnika). Dodatkowo każda transmisja pakietu może zakończyć się błędem TER. Brak wiadomości ACK po czasie (CTPk+ CTIZ) od wysłania pakietu jest dla stacji nadawczej sygnałem o konieczności retransmisji pakietu. Każdy pakiet może być retransmitowany maksymalnie LR razy. Dostęp do łącza w przypadku retransmisji opiera się na tych samych zasadach co transmisja pierwotna. Jeśli mimo LR-krotnej próby retransmisji pakietu nie udało się poprawnie odebrać, wówczas stacja nadawcza odrzuca pakiet i – jeśli jej bufor nie jest pusty – przystępuje do próby transmisji kolejnego pakietu. Opracuj symulator sieci bezprzewodowej zgodnie z metodą M.

Za pomocą symulacji wyznacz:

- Wartość parametru L, która zapewni średnią pakietową stopę błędów (uśrednioną po K odbiornikach) nie większą niż 0.1, a następnie:
  - o pakietową stopę błędów w każdym z odbiorników mierzoną jako iloraz liczby pakietów straconych do liczby przesłanych pakietów,
  - o średnią liczbę retransmisji pakietów, o przepływność systemu mierzoną liczbą poprawnie odebranych pakietów w jednostce czasu,
  - o średnie opóźnienie pakietu, tzn. czas jaki upływa między pojawieniem się pakietu w buforze, a jego poprawnym odebraniem,
  - o średni czas oczekiwania, tzn. czas między pojawieniem się pakietu w buforze, a jego opuszczeniem
- o Sporządź wykres zależności średniej liczby retransmisji pakietów od parametru P Sporządź wykres zależności przepływności systemu oraz średniej i maksymalnej pakietowej stopy błędów w zależności od wartości L.

<b>A3</b>	<p>Protokół CSMA (ang. <i>Carrier Sense Multiple Access</i>) z wymuszaniem transmisji z prawdopodobieństwem 1 (ang. <i>1-persistent</i>) – po wygenerowaniu nowego pakietu, stacja nadawcza sprawdza zajętość kanału transmisyjnego. Jeśli kanał jest zajęty, to czeka aż się zwolni sprawdzając zajętość co 0.5 ms. Jeśli kanał jest wolny, to stacja natychmiast podejmuje próbę przesłania swojego pakietu.</p> <p>W przypadku retransmisji, stacja nadawcza sprawdza stan kanału po losowym czasie CRP równym <math>R \cdot CTP_k</math>, gdzie R jest losową liczbą z przedziału od <math>&lt;0, (2^r - 1)&gt;</math>, a r jest numerem aktualnej retransmisji (przy każdej retransmisji czas ten jest losowany ponownie). Jeśli kanał jest zajęty, to czeka aż się zwolni, sprawdzając zajętość co 0.5 ms. Jeśli kanał jest wolny, to stacja natychmiast podejmuje próbę retransmisji swojego pakietu.</p>
-----------	--

Schemat modelu symulacyjnego:



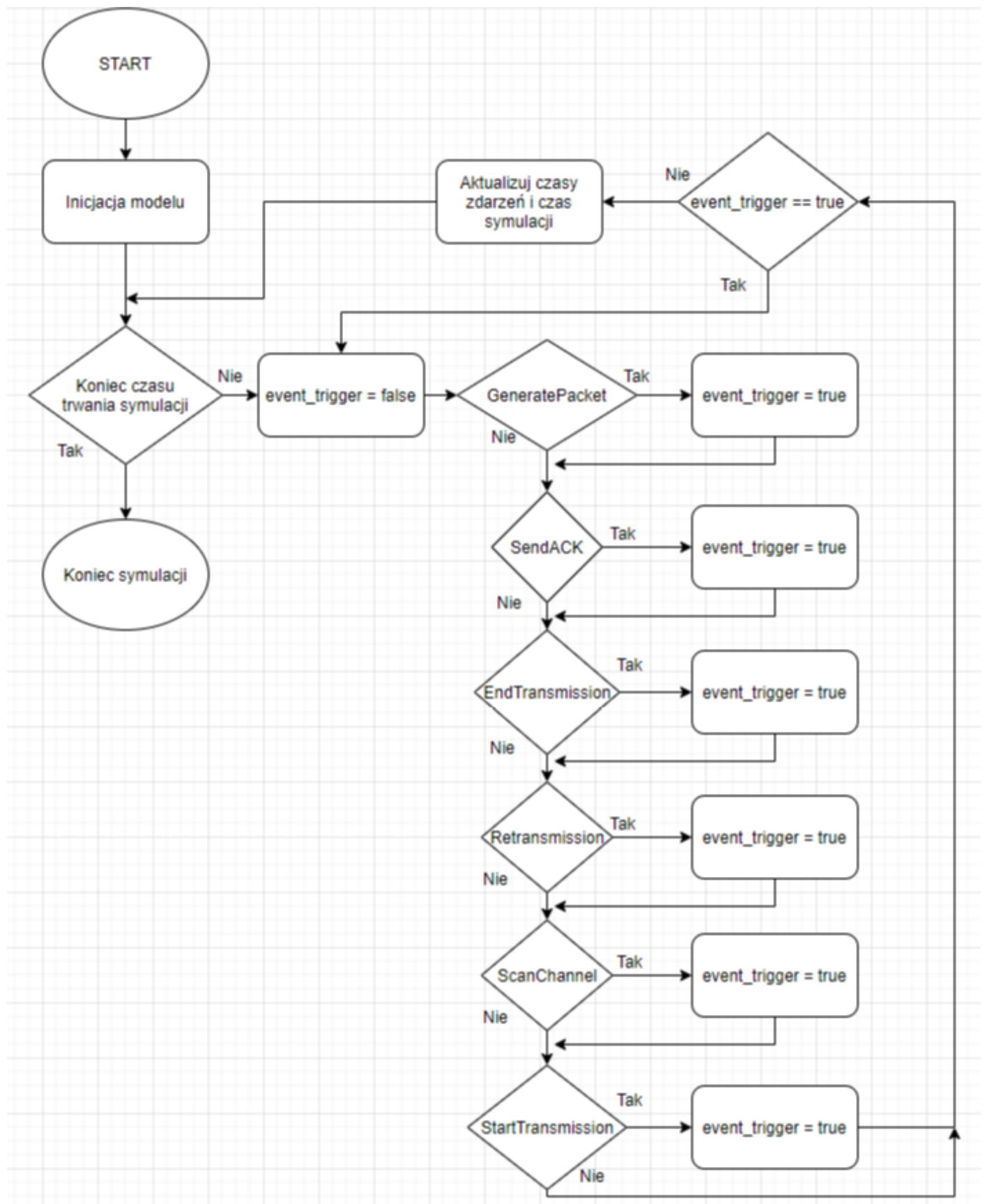
Opis klas wchodzących w skład systemu i ich atrybutów:

Obiekt	Nazwa klasy implementującej obiekt	Opis	Atrybuty
Sieć bezprzewodowa	WirelessNetwork	Klasa gromadząca wszystkie pozostałe elementy systemu.	<ul style="list-style-type: none"> <li>- k liczba stacji nadawczych i odbiorczych typu <i>const int</i></li> <li>- wektor stacji nadawczych <i>vector&lt;Transmitter*&gt;</i></li> <li>- wektor stacji odbiorczych <i>vector&lt;Receiver*&gt;</i></li> <li>CTIZ – czas transmisji ACK typu <i>const int</i></li> <li>- LR wartość maksymalnej możliwej liczby retransmisji typu <i>const int</i></li> </ul>
Stacja nadawcza	Transmitter	Klasa reprezentuje stację nadawczą, są w niej generowane pakiety w czasie CGPk, które umieszczane są w jej buforze w przypadku, gdy łącze jest zajęte i transmisja jest niemożliwa. Sprawdza ona czy dostęp do kanału jest możliwy, jeśli tak wysyła najstarszy pakiet, po czym czeka na odpowiedź ACK przez CTPK + CTIZ, jeśli jej nie otrzyma podejmowana jest próba retransmisji pakietu po czasie CPR, pod warunkiem że nie zostało przekroczone LR. W innym wypadku pakiet jest usuwany.	<ul style="list-style-type: none"> <li>- CGPk czas generowania kolejnego pakietu w k-tej stacji nadawczej typu <i>int</i></li> <li>- lista pakietów do wysłania, bufor <i>queue&lt;Packets*&gt;</i></li> <li>- czas oczekiwania na odpowiedź (CTPk + CTIZ) typu <i>int</i></li> <li>- czas sprawdzania zajętości kanału przez stację nadawczą (0.5ms) typu <i>const int</i></li> <li>- CPR czas po którym sprawdzany jest stan kanału w przypadku retransmisji typu <i>int</i></li> <li>-id stacji nadawczej typu <i>int</i></li> </ul>
Stacja odbiorcza	Receiver	Klasa reprezentuje stację odbiorczą, w której nadawana jest wiadomość o pomyślnym zakończeniu transmisji ACK, oraz sprawdzane czy nie wystąpił błąd TER.	<ul style="list-style-type: none"> <li>- Zmienna typu <i>bool</i> informująca o nadaniu komunikatu ACK.</li> <li>- Zmienna typu <i>bool</i> informująca o błędzie typu TER</li> </ul>
Kanał transmisji (łącze)	Channel	Klasa reprezentują łącze, w którym odbywa się transmisja. Przechowuje informacje o dostępności kanału. Posiada wektor w którym znajdują się pakiety, które zostały nadane w momencie sprawdzenia, że kanał jest dostępny. Znajduje się tu także informacja o tym, czy wystąpiła kolizja.	<ul style="list-style-type: none"> <li>- Zmienna informująca o dostępności kanału typu <i>bool</i> ustawione na false, podczas trwania CTPk + CTIZ</li> <li>- wektor przechowujący nadane pakiety <i>vector&lt;Packets*&gt;</i>, gdyż może się zdarzyć, że dwa będą nadane w tym samym czasie.</li> <li>- zmienna typu <i>bool</i> informująca o kolizji pakietów, jeśli w wektorze umieszczone zostanie więcej niż jeden pakiet</li> </ul>
Pakiety	Packets	Klasa reprezentuje pakiety o czasie transmisji CTPKk, przechowuje też informacje o liczbie retransmisji.	<ul style="list-style-type: none"> <li>- CTPk czas transmisji pakietu typu <i>int</i></li> <li>- r numer aktualnej retransmisji typu <i>int</i></li> </ul>

Krótki opis przydzielonej metody symulacyjnej:

W metodzie tej wszystkie zdarzenia zawarte są w pętli głównej. Należy sprawdzać w odpowiedniej kolejności wszystkie warunki zdarzeń podczas iteracji tej pętli. Po spełnieniu warunku danego zdarzenia wywoływane jest zdarzenie i ustawia się flaga event\_trigger w celu ponownego wywołania tej samej iteracji pętli, aby przykładowo nie pominąć żadnego ze zdarzeń w danej chwili czasowej. Po wykonaniu się wszystkich zdarzeń i braku aktywnej flagi event\_trigger można przejść do aktualizacji zegarów symulacji.

Schemat blokowy pętli głównej:



**Lista zdarzeń czasowych i warunkowych:**

Zdarzenie	Opis	Algorytm
Wygenerowanie pakietu (czasowe)	Pakiet jest generowany w k-tej stacji nadawczej w czasie CGPk i umieszczany w buforze. W czasie tego zdarzenia generowany jest czas pojawienia się kolejnego pakietu.	1)Umieść pakiet na końcu bufora. 2)Wygeneruj czas pojawienia się nowego pakietu.
Sprawdzenie dostępności łącza (czasowe)	Po wygenerowaniu pakietu, jeśli nie ma żadnych pakietów w buforze stacja nadawcza sprawdza dostępność łącza, jeśli jest niedostępne, to powtarza tę czynność po czasie 0.5ms, dopóki bufor się nie zwolni, bądź dopóki nie uzyska dostępu do łącza.	1)Sprawdź czy łącze jest dostępne. 2)Jeśli nie, poczekaj 0.5ms. 3)Powtórz.
Rozpoczęcie transmisji (warunkowe)	Po uzyskaniu dostępu do łącza k-ty terminal podejmuje próbę transmisji. Pakiet 'opuszcza' bufor i rozpoczyna się oczekiwanie na odpowiedź ACK, trwające (CTPk + CTIZ). Ustawiamy czas zajętości na czas CTPk i sprawdzamy, czy wystąpiła kolizja, lub czy pojawił się błąd TER. Będzie to miało znaczenie dla czasu zajęcia łącza.	1)Jeśli łącze jest wolne: a) Ustaw flagę dostępności kanału na zajęty na czas CTPk. b)Dodaj pakiet do wektora pakietów w łączu. c)Ustaw czas oczekiwania na odpowiedź na (CTPk + CTIZ). 2)Sprawdź czy wystąpiła kolizja pakietów i ustaw true jeśli tak w zmiennej bool. 3)Sprawdź, czy wystąpił błąd TER i jeśli tak ustaw true.
Transmisja ACK (warunkowe)	Sprawdzenie czy nie wystąpiła kolizja lub nie pojawił się błąd typu TER i nadanie odpowiedzi ACK oraz wyczyszczenie kolejki w łączu.	1)Jeśli transmisja przebiegła pomyślnie bez błędu lub kolizji: a)Flaga dostępności kanału pozostaje zajęta na dodatkowy czas CTIZ. B)Zostaje wysłana odpowiedź ACK. C)Pakiet zostaje usunięty z wektora w łączu. 2)Jeśli nie przebiegła pomyślnie: a)Zwolnij flagę dostępności łącza. b)Wyczyść zawartość wektora pakietów.
Zakończenie transmisji (warunkowe)	Jeśli pojawia się odpowiedź ACK, transmisja zostaje zakończona.	1)Jeśli pojawiła się odpowiedź ACK: a)Usuń pakiet z bufora. b)Podejmij próbę transmisji kolejnego pakietu.
Retransmisja (warunkowe)	Jeśli po czasie(CTPk + CTIZ) nie pojawiła się odpowiedź ACK, podejmowana jest decyzja o retransmisji. W przypadku retransmisji dla każdej kolejnej wyznaczany jest nowy czas oczekiwania, gdyż wartość CRP zależy od zmiennej r.	1)Jeśli liczba retransmisji jest mniejsza lub równa od LR. a)Inkrementuj liczbę transmisji. b)Wygeneruj i poczekaj czas CRP. c) Sprawdzenie dostępności łącza. 2)Jeśli nie, odrzuć pakiet: a)usuń pakiet z bufora. b)Jeśli bufor nie jest pusty, podejmij próbę transmisji kolejnego pakietu.

**Parametry wywołania programu:**

Lambda = 0.0006655

Liczba symulacji = 10

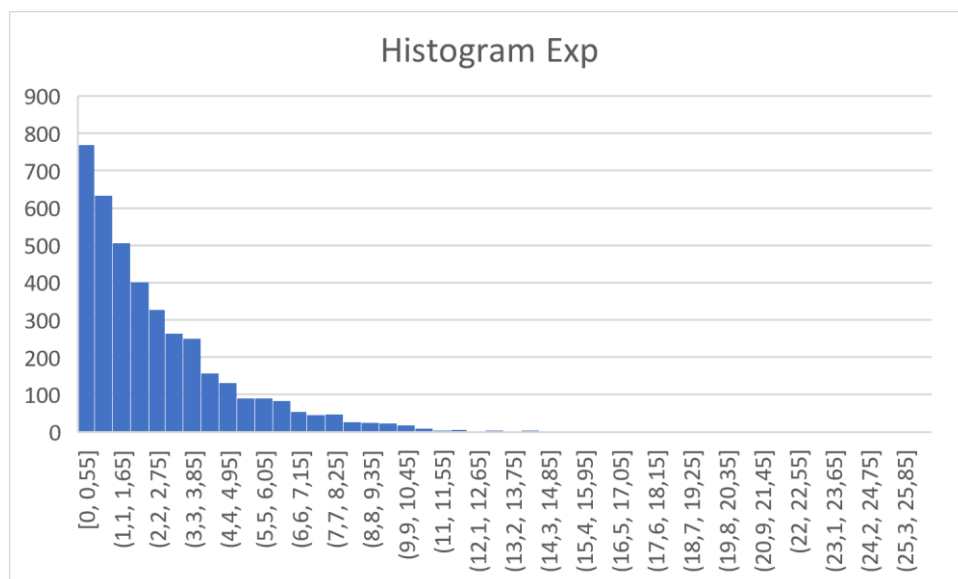
Czas symulacji = 400000 – 400000 \* 0.1ms = 40s

Faza początkowa = 320 – liczba transmisji pakietów zakończonych sukcesem

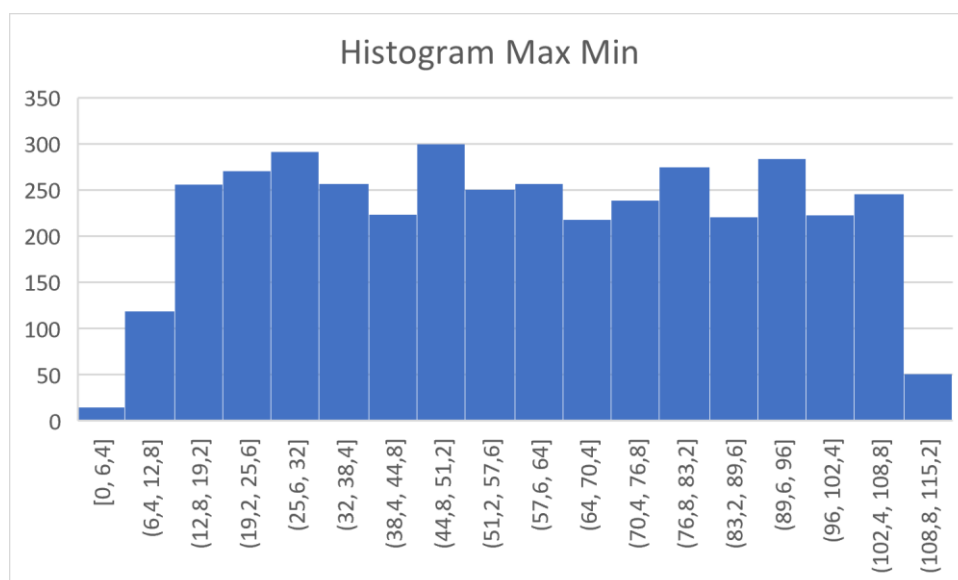
Błąd TER = 0.2

Parametry te wprowadzane są wewnątrz kodu, wcześniej można było wybrać jeszcze tryb krokowy, ale w celu pobrania poszczególnych wyników program został bardziej zautomatyzowany.

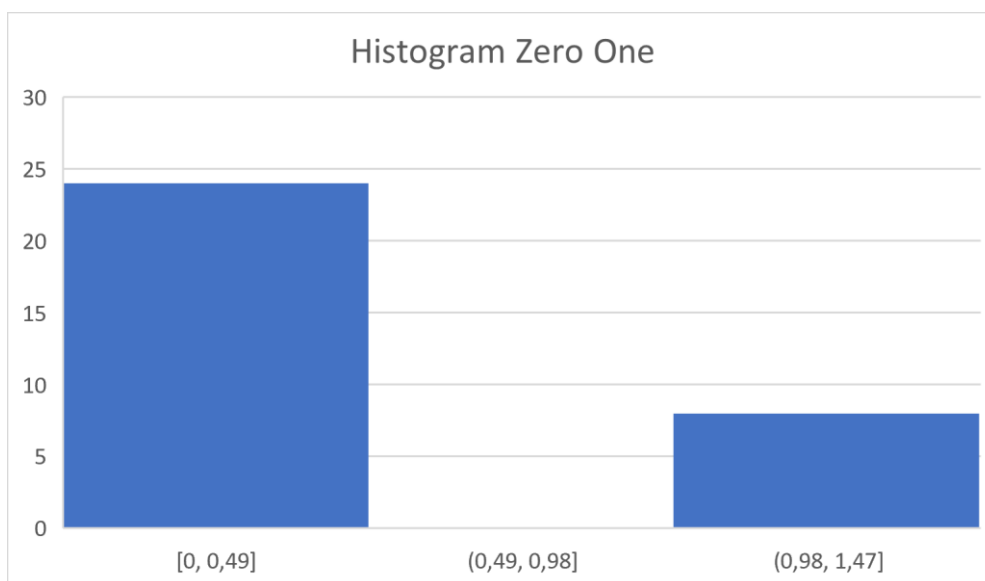
**Generatory:**  
Rozkład wykładniczy:



Rozkład jednostajny:



Rozkład zero jedynkowy:



#### Sposób zapewnienia niezależności generatorów:

W każdej symulacji losuje 31 ziaren (10 CGP 10 CTP 10 CRP + 1 TER) z odstępem co 10000 losowań. Odbywa się to przed rozpoczęciem pętli głównej, poszczególne ziarna trafiają do wektorów należących do klasy WirelessNetwork w funkcji Initialization przy tworzeniu stacji nadawczych.

```
for(int i = 0; i < k_number_ ; i++)
{
    //Building generators
    UniformGenerator* temp_gen = new UniformGenerator(root->generate_kernel(k:100000));
    CGP_kernel_vector_.push_back(temp_gen);
    temp_gen = new UniformGenerator(root->generate_kernel(k:100000));
    CTP_kernel_vector_.push_back(temp_gen);
    temp_gen = new UniformGenerator(root->generate_kernel(k:100000));
    CRP_kernel_vector_.push_back(temp_gen);
    root->set_kernel(temp_gen->get_kernel());
}
```

Po tej pętli tworzę generator nr 31 dla TER:

```
UniformGenerator* temp_gen = new UniformGenerator(root->generate_kernel(k:100000));
TER_kernel = temp_gen;
root->set_kernel(temp_gen->get_kernel());
```

Przy okazji zapisuje wartość generatora źródłowego w odległości 100000 od ostatniego wygenerowanego ziarna. Z racji tego, że Root wywoływany jest w main'ie i przekazywany z konstruktorem klasy Simulator, jest zachowana niezależność generatorów w kolejnych symulacjach.

```
UniformGenerator* root = new UniformGenerator(kernel:123);
Simulator* test_simulator = new Simulator(root);
for(int i = 0 ; i < 10 ; i++)
{
    test_simulator->simulation(time:400000, &logger, sym_numb:i+1, lambda:0.0006655);
}

return 0;
```

#### Krótki opis zastosowanej metody testowania i weryfikacji poprawności działania programu:

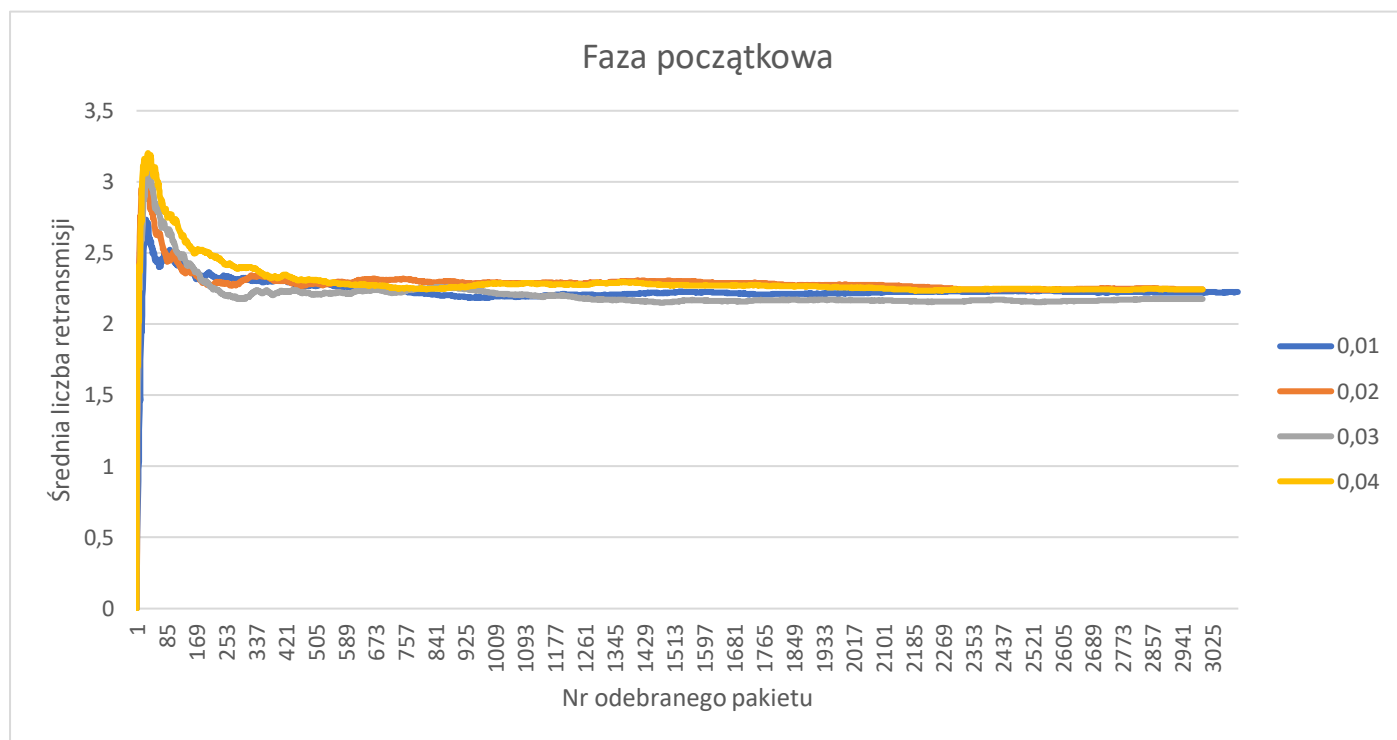
Testowanie aplikacji odbywało się głównie za pomocą logger'a. Początkowo starałem się wyświetlać każde zdarzenie bardzo szczegółowo, prześwietlałem każdą generowaną zmienną wraz z danymi obiektu do którego należy. Pozwoliło mi to stworzyć szkielet symulacji, lecz nie uchroniło mnie przed błędami, które pojawiły się po prawidłowym zaimplementowaniu generatorów. Wtedy niezbędny okazał się debugger, który umożliwił zlokalizowanie problemu.

#### Wyniki symulacji:

Aby uzyskać poniższe wyniki zmuszony byłem do wprowadzeniu pewnych zmian w założeniach początkowych projektu:

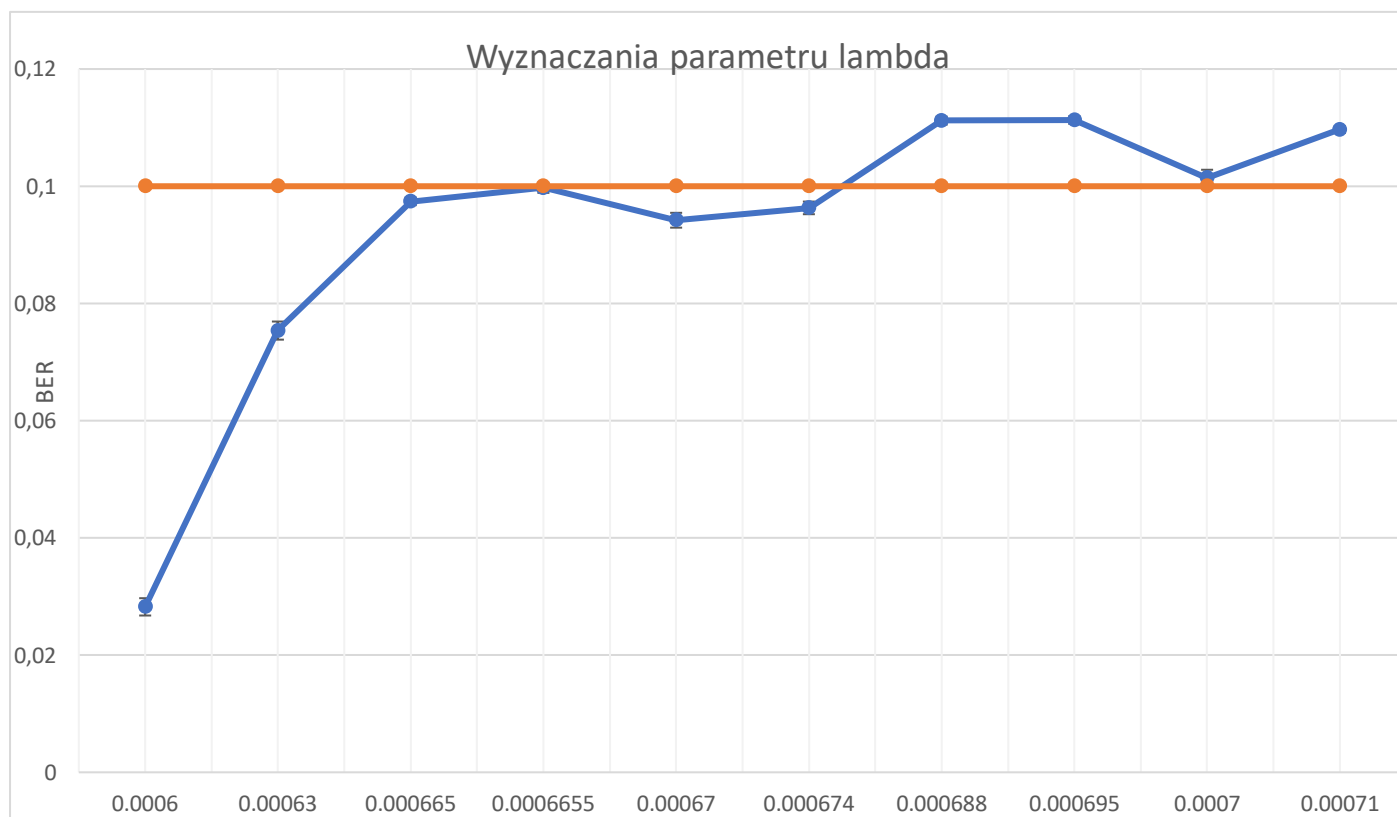
- zmiana wartości CRP z  $R \cdot CTP$  na R
- zmniejszenie wartości dopuszczalnej liczby retransmisji z 15 na 12

A) Wyznaczanie długości fazy początkowej:



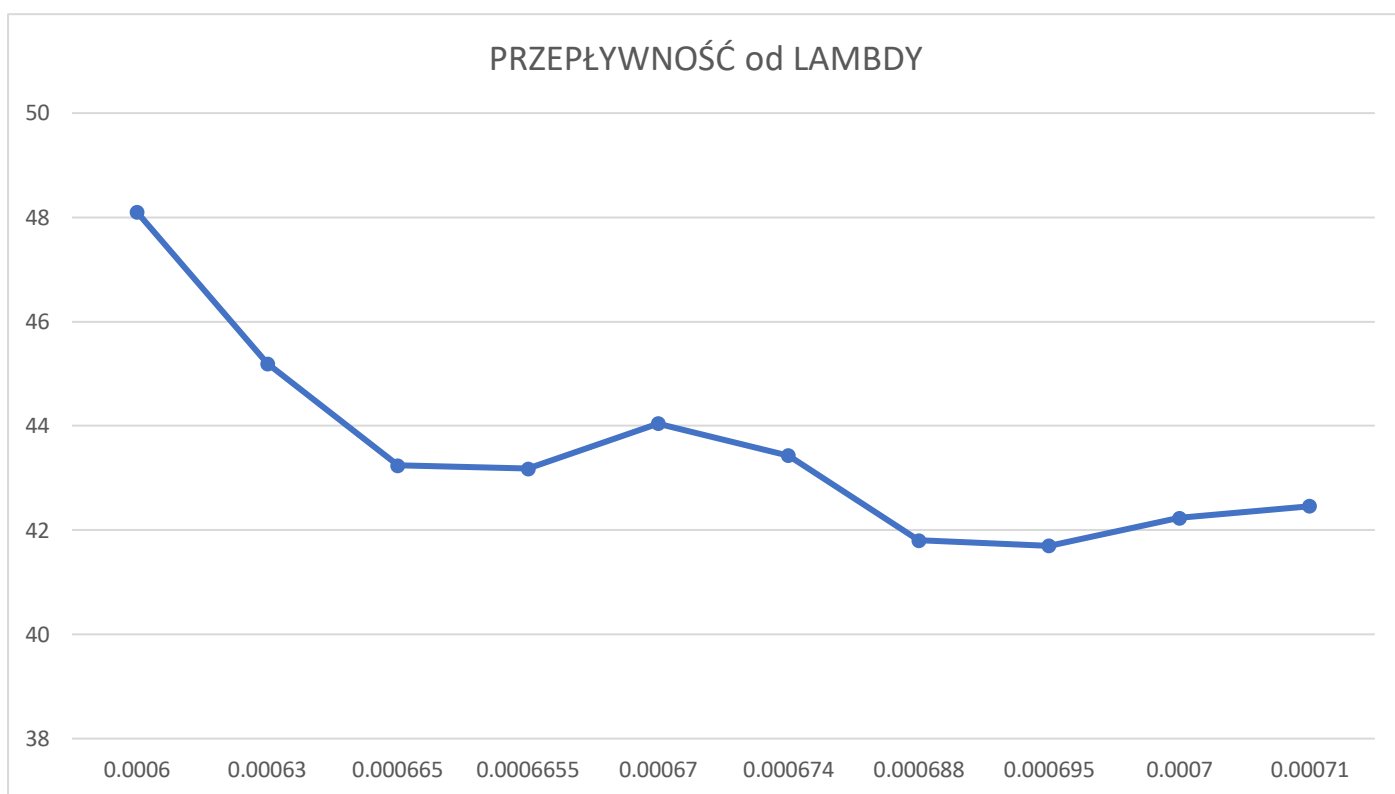
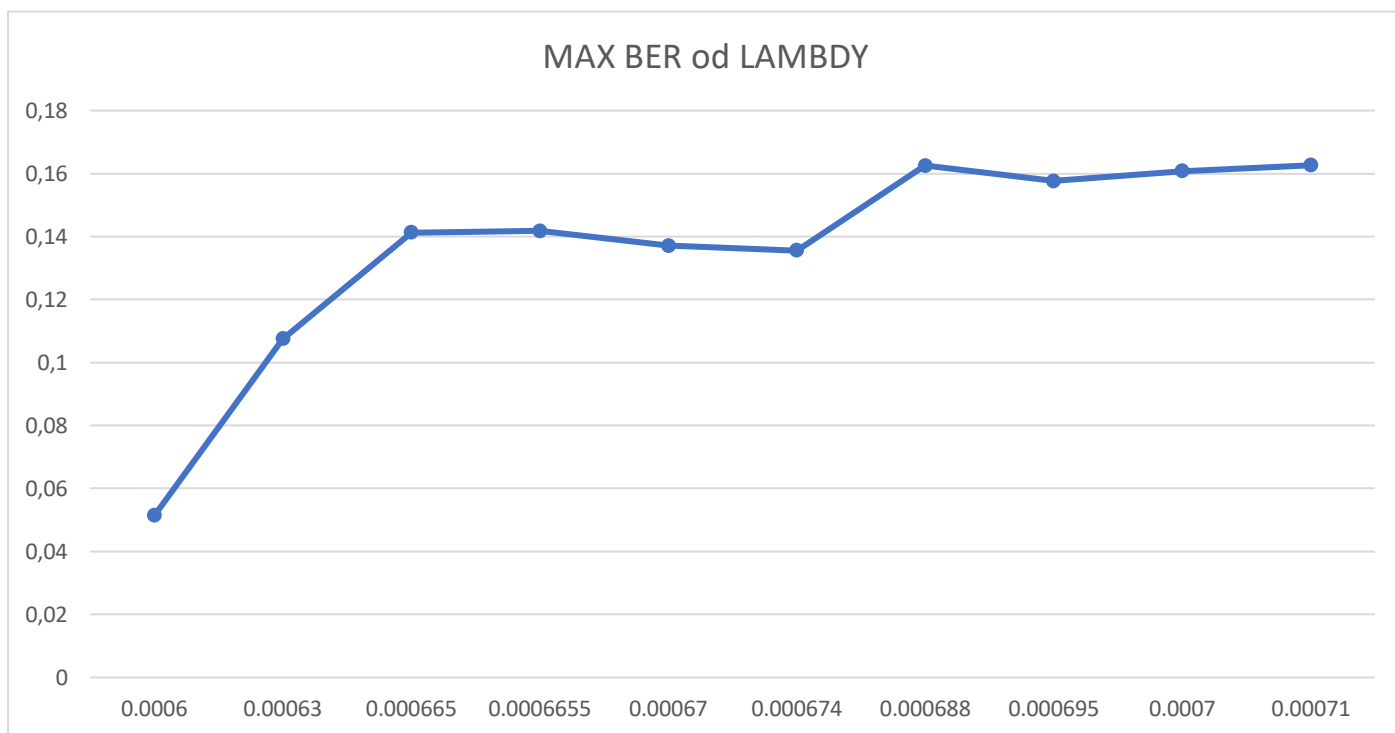
Przeprowadziłem 10 symulacji dla każdej wartości parametru  $\lambda$  i wyniki te uśredniłem. Uznałem, że najlepszą metodą wyznaczenia fazy początkowej będzie wykres średniej liczby retransmisji na dany moment wyrażony w wartości odebranych pakietów. Uznałem, że wykres uzyskuje stabilność dla wartości około **320** odebranych pakietów.

B) Wyznaczenie wartości parametru  $\lambda$ :



Wykonałem 10 symulacji dla różnych wartości parametru  $\lambda$  w celu uzyskania wartości TER jak najbardziej zbliżonej do 0,1, co udało mi się dla  $\lambda$  **0.0006655**.

C) Wykres maksymalnej pakietowej stopy błędów i przepływności w zależności od wartości parametru lambda:





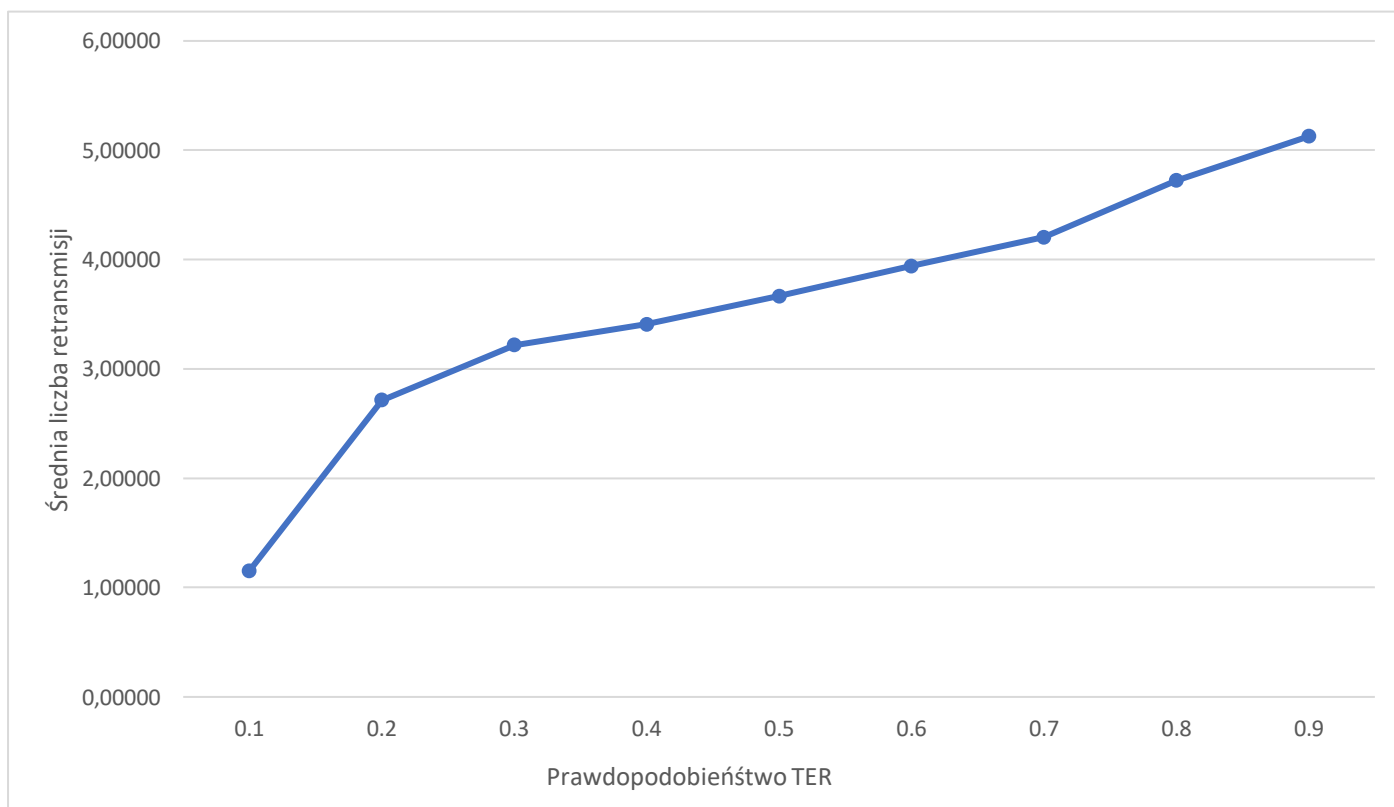
D) Tabelka z wynikami symulacji:

Nr symulacji	Średni BER	Max BER	Średnia LR	Średni czas oczekiwania [ms]	Średnie opóźnienie [ms]	Przepływność
1	0,0603503	0,096154	2,11452	277,955	1429,55	46,5
2	0,109346	0,152	2,90621	468,218	2873,88	41,85
3	0,0750021	0,104294	2,32428	366,718	1890,55	45,1
4	0,105445	0,198198	2,91724	577,264	3227,83	43,8
5	0,0836384	0,107143	2,81278	463,482	2714,23	44,2
6	0,111185	0,181159	2,91043	897,136	2865,19	42,425
7	0,115523	0,150376	3,05286	237,455	4786,94	39,725
8	0,122295	0,170543	3,05912	283,582	4162,78	40,175
9	0,114706	0,170543	3,06404	820,591	4715,58	40,6
10	0,0632753	0,086705	1,96888	353,109	1491,32	47,4
Średnia:	0,0960766	0,141712	2,713036	474,551	3015,785	43,1775
Odchylenie:	0,023228	0,039842	0,4149308	227,728915	1230,487429	2,664672431
Ufność:	0,000923	0,001583	0,0164884	9,049402124	48,89662584	0,105887706

E) Wyniki końcowe:

$$\begin{aligned}
 \Lambda &= 0,0006655 \\
 \text{Średni BER} &= 0,00960766 \pm 0,000923 \\
 \text{Max BER} &= 0,141712 \pm 0,001583 \\
 \text{Średnia LR} &= 2,713036 \pm 0,0164884 \\
 \text{Średni czas oczekiwania} &= 227,7728915 \pm 9,049402124 \\
 \text{Średnie opóźnienie} &= 1230,487429 \pm 48,89662584 \\
 \text{Przepływność} &= 43,1775 \pm 0,105887706
 \end{aligned}$$

F) Wykres zależności średniej liczby retransmisji pakietów od parametru P:



**Wnioski:**

- Dla mniejszych wartości parametru  $\lambda$  system generował mniej zdarzeń, natomiast po przekroczeniu pewnej wartości był na tyle zapełniony, że nie miało to większego znaczenia dla wyników. Przez to początkowo miałem problem z dobraniem odpowiedniej  $\lambda$ , gdyż nie spodziewałem się, że uzyskam oczekiwany rezultat dla tak małych wartości tego parametru.
- $\lambda$  na których operowałem prawdopodobnie są zbyt duże, aby uzyskać sensowny przebieg, ale mimo tego udało się znaleźć wartość odpowiadającą potrzebom symulacji.
- Założenia początkowe zapewniały zbyt korzystne warunki dla systemu sieci bezprzewodowej przez co byłem zmuszony je zmodyfikować w celu zwiększenia wartości BER. Cel ten był możliwy do zrealizowania poprzez zmniejszenie liczby retransmisji, modyfikację wzoru generatorów, a także opcja z której nie skorzystałem, czyli zwiększenie liczby stacji nadawczych, co zwiększyłoby kolejkę i pomogło uzyskać oczekiwany rezultat.