

## Symulacja cyfrowa

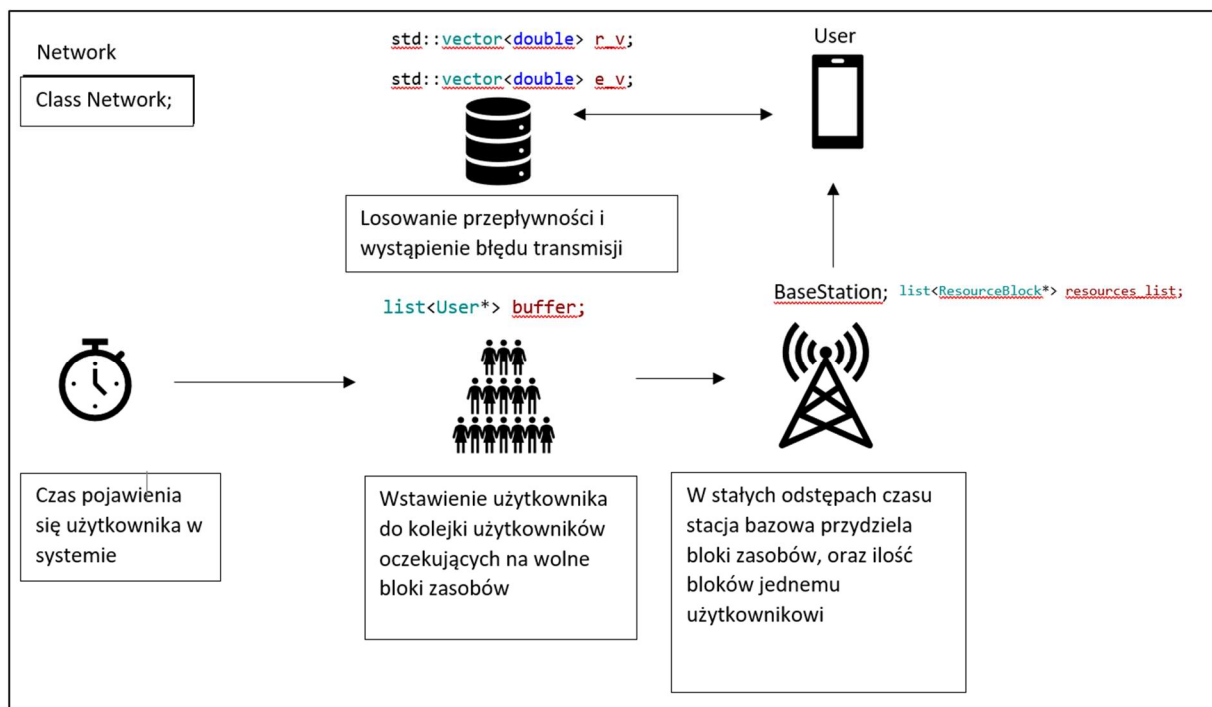
# Raport końcowy

### 1. Imię, nazwisko, nr zadania i nazwa metody symulacyjnej:

Jakub Rybski, zad 5, M2A1b

### 2. Pełny tekst rozwiązywanego zadania 3. Krótki opis modelu symulacyjnego:

#### a. schemat modelu symulacyjnego



#### b. opis klas wchodzących w skład systemu i ich atrybutów

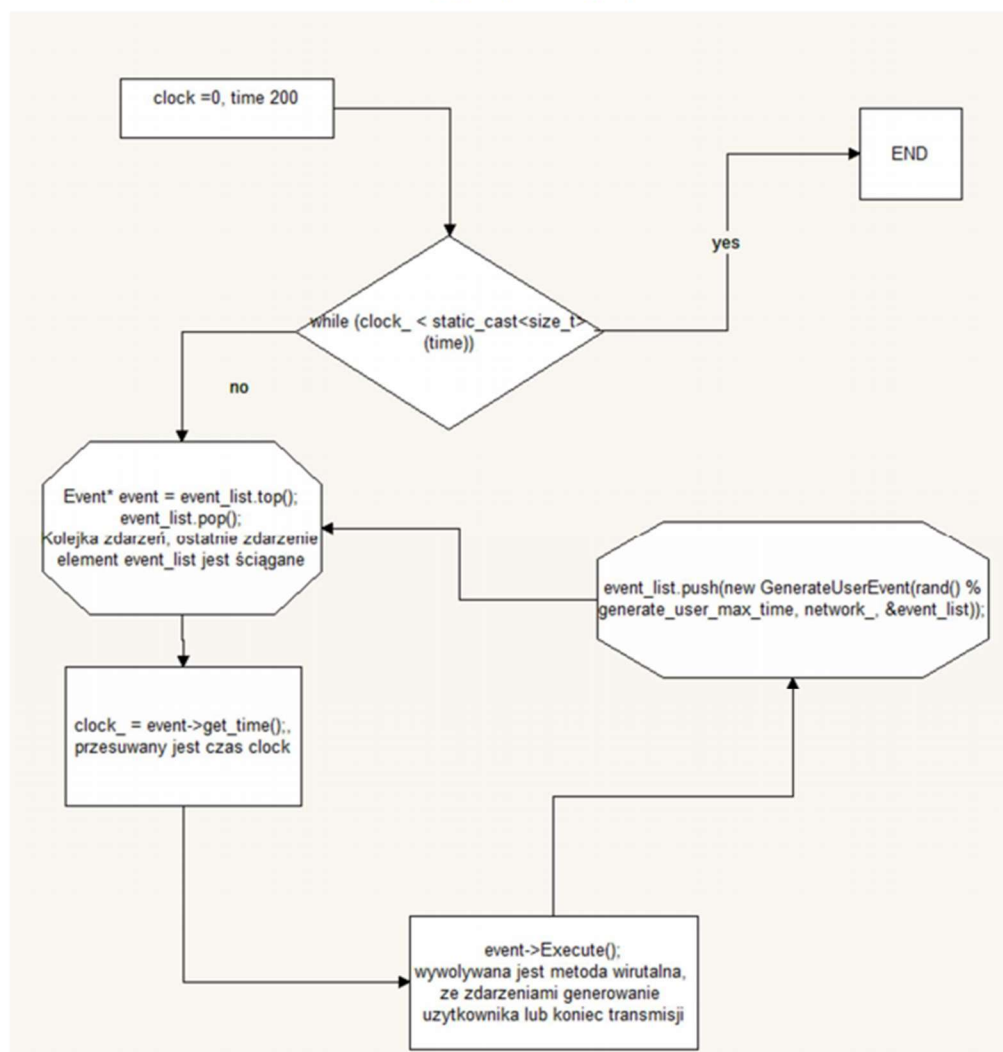
Obiekt	Nazwa klasy implementującej obiekt	Opis	Najważniejsze atrybuty
Network	Network	Klasa opisująca pojedynczą komórkę sieci mobilnej, gromadzi pozostałe elementy systemu.	<ul style="list-style-type: none"> <li>- kolejka użytkowników oczekujących na wolne bloki zasobów (kolejka typu &lt;User*&gt;)</li> <li>- wskaźnik na stację bazową</li> </ul>

Base Station	BaseStation	Klasa opisująca stację bazową, w chwili pojawienia się użytkownika stacja bazowa ustala warunki propagacyjne oraz przepływność. W stałych odstępach czasu $s$ stacja bazowa przydziela użytkownikom bloki zasobów.	<ul style="list-style-type: none"> <li>- liczba bloków <math>k</math> zasobów (typu const int)</li> <li>- zmienna <math>s</math> która reprezentuje stałe odstępy czasu w których stacja bazowa przydziela bloki zasobów (typu const int)</li> <li>- zmienna <math>l</math> reprezentująca ilość bloków zasobów które można przydzielić jednemu użytkownikowi (typu const int)</li> <li>-lista bloków zasobów list &lt;ResourceBlock*&gt;</li> </ul>
Użytkownik	User	Klasa opisująca pojedynczego użytkownika.	<ul style="list-style-type: none"> <li>- wektor losowych przepływności typu int</li> <li>- wektor losowych błędów o typu int</li> <li>-czas przybycia typu int</li> <li>-wskaźniki na generatory: danych, przepływności i błędów</li> </ul>
Symulator	Simulator	Klasa z symulacją realizująca symulacje M2	-lista zdarzeń
Blok zasobu	ResourceBlock	Klasa bloku zasobu	<ul style="list-style-type: none"> <li>-zmienna reprezentująca id bloku</li> <li>-zmienna reprezentująca przepływność użytkownika który używa bloku</li> <li>-zmienna reprezentująca błąd transmisji użytkownika który używa bloku</li> </ul>
Klasa zdarzenia	AllocateUser	Klasa zdarzenia przydziału zasobów	-wskaźniki na generatory: czasu, tau
Klasa zdarzenia	ChangeRk	Klasa zdarzenia zmiany przepływności użytkownika	-wskaźniki na generatory: czasu, tau
Klasa generator	Generator	Klasa z generatorami	-ziarno typu int
Klasa zdarzenia	Event	Klasa z której dziedziczą inne zdarzenia	
Klasa zdarzenia	GenerateUserEvent	Klasa generująca użytkowników	

### 3. Opis przydzielonej metody symulacyjnej

#### a. Schemat blokowy pętli głównej

*Schemat pętli symulacyjnej*



#### b. Lista zdarzeń czasowych/warunkowych lub zidentyfikowane procesy (z zad. 1 – zaktualizowane)

Zdarzenie	Warunek Czasowy	Algorytm
Pojawienie się użytkownika	Upłynięcie czasu wylosowanego przez $t$ (czas do pojawienia się kolejnego użytkownika w obszarze komórki)	1. Umieść użytkownika do kolejki użytkowników oczekujących na wolne bloki zasobów
Zdarzenie przydziału zasobów	Upłynięcie czasu $s$	1. Przydzielenie bloków zasobów użytkownikowi, pierwszemu użytkownikowi jest przypisywany pierwszy

		blok zasobów, drugiemu użytkownikowi drugi blok zasobów itd., aż do wykorzystania wszystkich zasobów radiowych, lub przypisania każdemu użytkownikowi I bloków zasobów.
Zmiana warunków propagacyjnych	Upłynięcie czasu $\tau$	1. Ustalana jest nowa przepływność dla użytkownika oraz błąd transmisji

Zdarzenie	Warunek	Algorytm
Zakończenie obsługi użytkownika (całkowite)	Użytkownik odebrał wszystkie dane jakie miał do odebrania	1. Użytkownik jest usuwany z systemu
Błąd transmisji	Pojawienie się błędu podczas przesyłania bloku zasobu	1. Dane są przesyłane, aż do poprawnego odebrania wszystkich danych

#### 4. Parametry wywołania programu (lambda, nr symulacji, koniec fazy początkowej, itp.)

czas symulacji – czas trwania symulacji

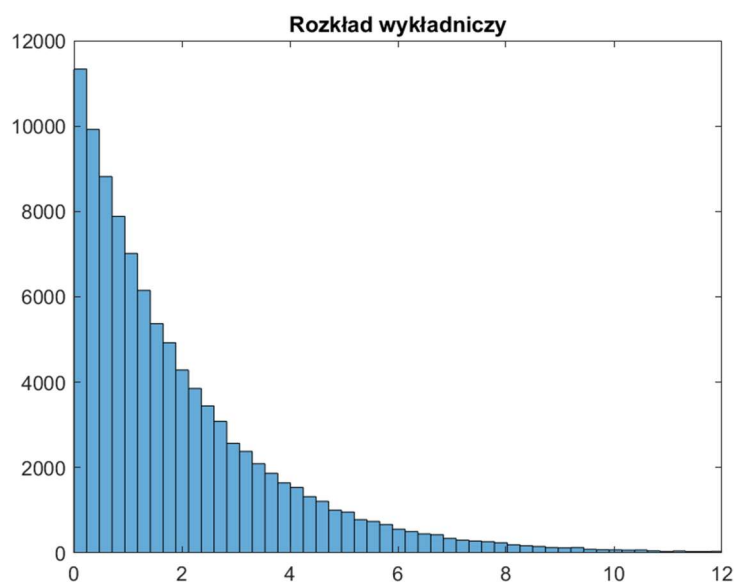
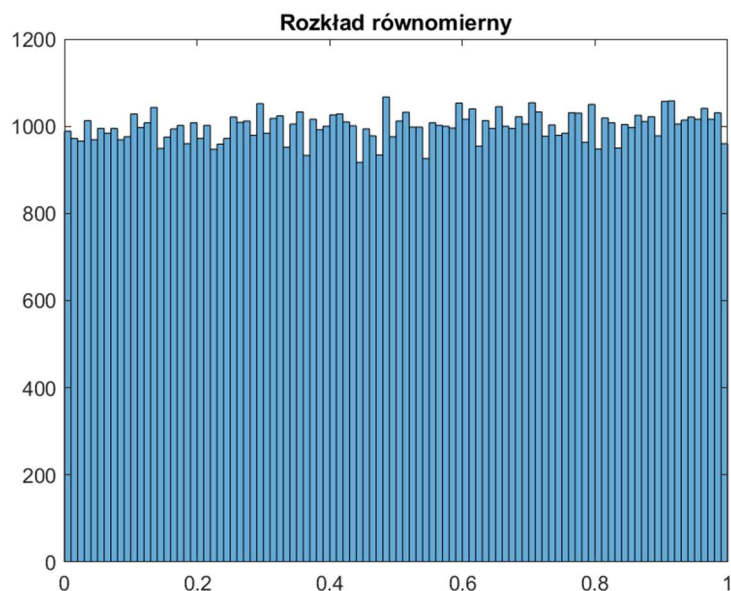
nr symulacji – jest numerem symulacji dla danego zestawu ziaren od 1 do 10.

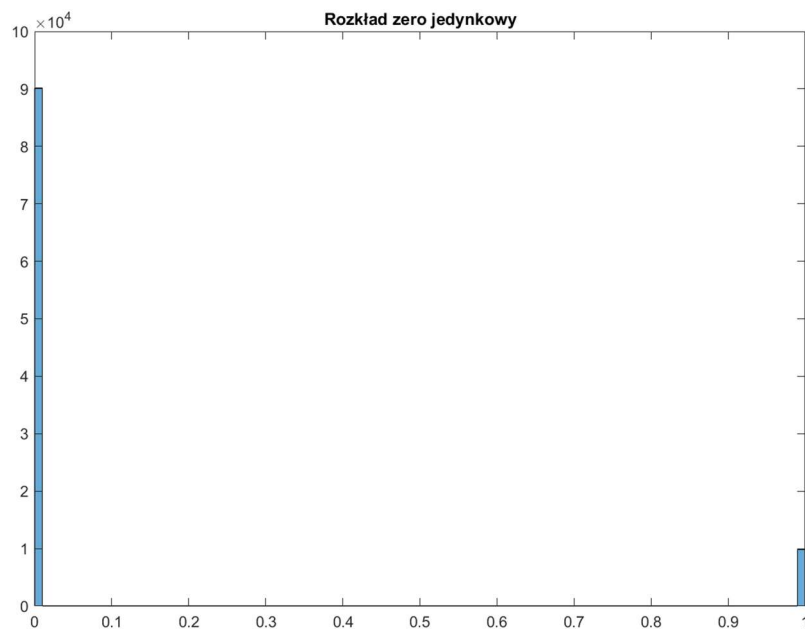
lambda – intensywność zgłoszeń nowych użytkowników w sieci

koniec fazy początkowej – czas po którym system się stabilizuje, oraz są zbierane statystyki

#### 5. Generatory

##### a. Opis zastosowanych generatorów liczb losowych z histogramami





***b. Wyjaśnienie, w jaki sposób została zapewniona niezależność sekwencji losowych w różnych symulacjach (jak wygenerowano ziarna, ile ziaren wygenerowano, gdzie są przechowywane, ile przewidziano symulacji)***

Niezależność sekwencji losowych została zapewniona poprzez użycie generatorów o rozkładzie:

- równomiernym,
- wykładniczym,
- jednostajnym,
- zero jedynkowym

Przewidziano 10 symulacji:

Zostało utworzonych 6 generatorów, 1 generator główny (rozkład równomierny) na podstawie którego zostały wygenerowane ziarna. Wygenerowano 50 ziaren z odstępem co 100 000 losowań, są one trzymane w pliku (odczyt z pliku za pomocą odpowiedniej metody). Następnie stworzono 5 pozostałych generatorów do zmiennych: (czasu, danych, przepływności, tau, epsilon). Na podstawie różnorodności ziaren i przy każdym losowaniu z pozostałych generatorów używania rozkładu równomiernego podając za każdym razem inne ziarno została zapewniona niezależność sekwencji losowych.

## 6. Krótki opis zastosowanej metody testowania i weryfikacji poprawności działania programu

Metoda, którą wybrałem, to porównanie symulacji z podobnym algorytmem

	Mój symulator	Symulator kolegi
Metoda	M2	M2
Algorytm	A1b	A1a
Liczba bloków zasobów	25	25
Maks. liczba bloków dla użytkownika	3	3
Czas pomiędzy przydziałami RB	5ms	5ms
Zestawy ziaren	Takie same	

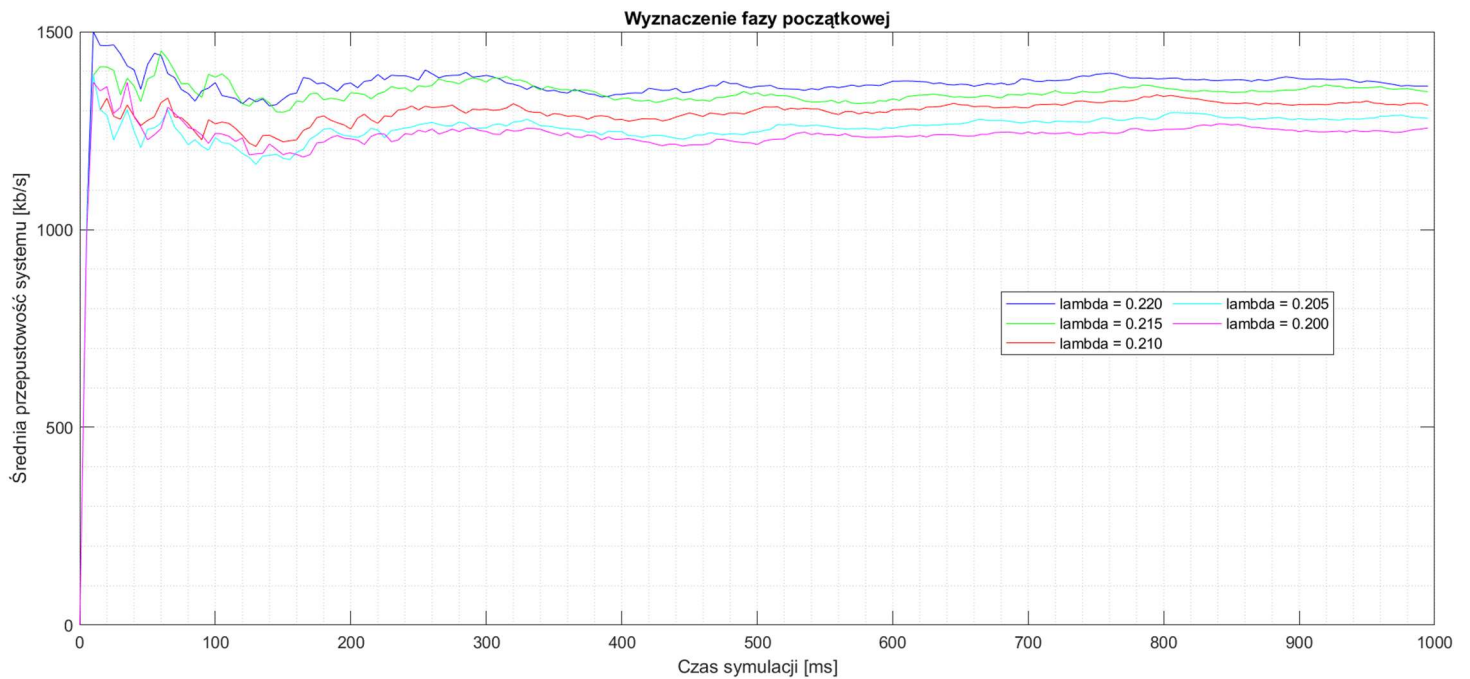
Nr symulacji (zestaw ziaren)	1	1
Lambda	1.00	1.00
Koniec fazy początkowej	500	500
Koniec symulacji	5000	5000
Uzyskane wyniki		
Średnia przepływność systemu [ $kb/s$ ]	7765	6213
Średnia przepływność użytkownika [ $kb/s$ ]	145	133
Czas oczekiwania [ $ms$ ]	8.87	10.34
Liczba wszystkich użytkowników	7933	4818

Nr symulacji (zestaw ziaren)	5	5
Lambda	1.3	1.3
Koniec fazy początkowej	500	500
Koniec symulacji	5000	5000
Uzyskane wyniki		
Średnia przepływność systemu [ $kb/s$ ]	8854	7793
Średnia przepływność użytkownika [ $kb/s$ ]	125	123.35
Czas oczekiwania [ $ms$ ]	10.74	11.48
Liczba wszystkich użytkowników	11806	6121

Nr symulacji (zestaw ziaren)	7	7
Lambda	1.5	1.5
Koniec fazy początkowej	500	500
Koniec symulacji	5000	5000
Uzyskane wyniki		
Średnia przepływność systemu [ $kb/s$ ]	9246	8902
Średnia przepływność użytkownika [ $kb/s$ ]	9	39.17
Czas oczekiwania [ $ms$ ]	159.97	44.37
Liczba wszystkich użytkowników	15968	7485

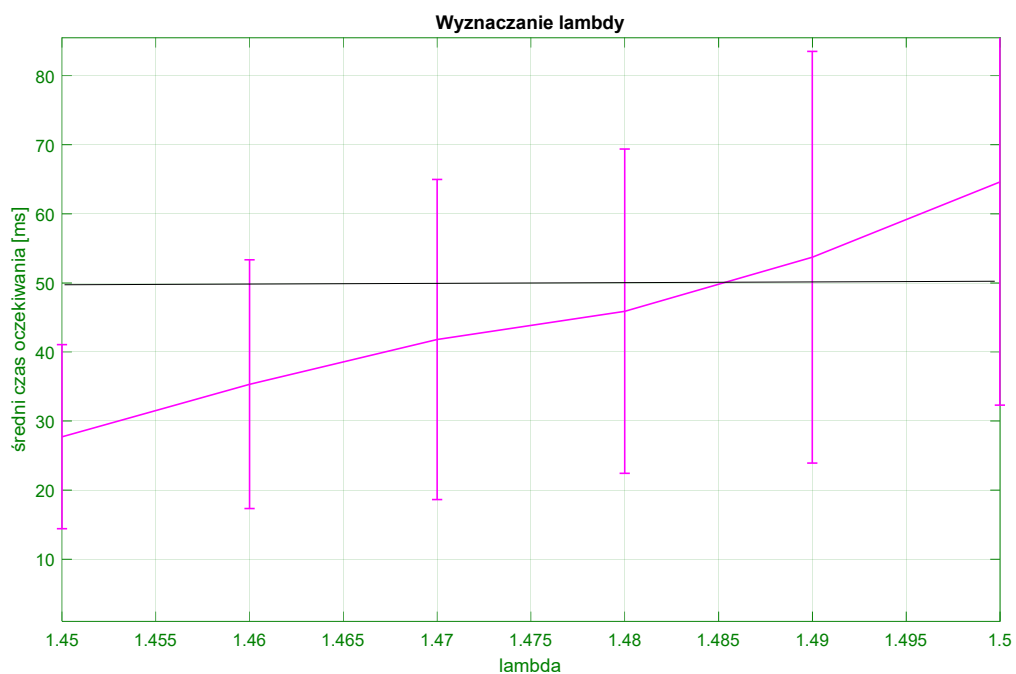
## 7. Wyniki symulacji

### a. Wyznaczenie długości fazy początkowej



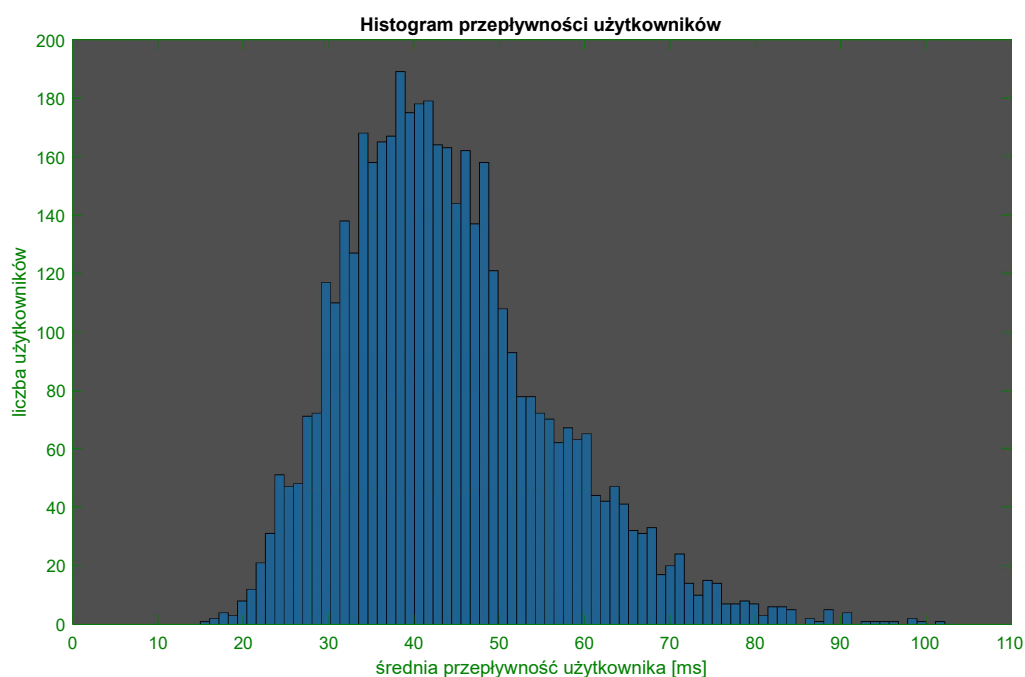
Za moment końca fazy początkowej uznałem czas 500ms, ponieważ po nim system się stabilizuje.

### b. Wyznaczenie wartości parametru lambda (wykres z przedziałami ufności i opis)



Przedziały ufności dla lambda 1.485: [23.63 , 77.45]

**c. Histogram przepływności użytkowników (uśredniony po wszystkich przebiegach symulacyjnych)**



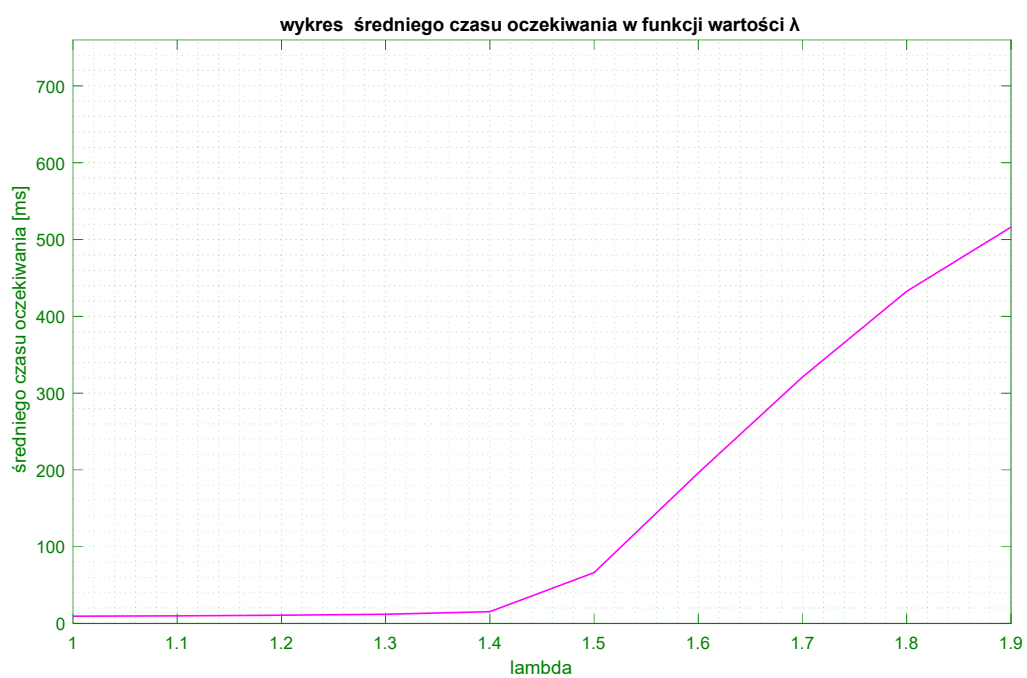
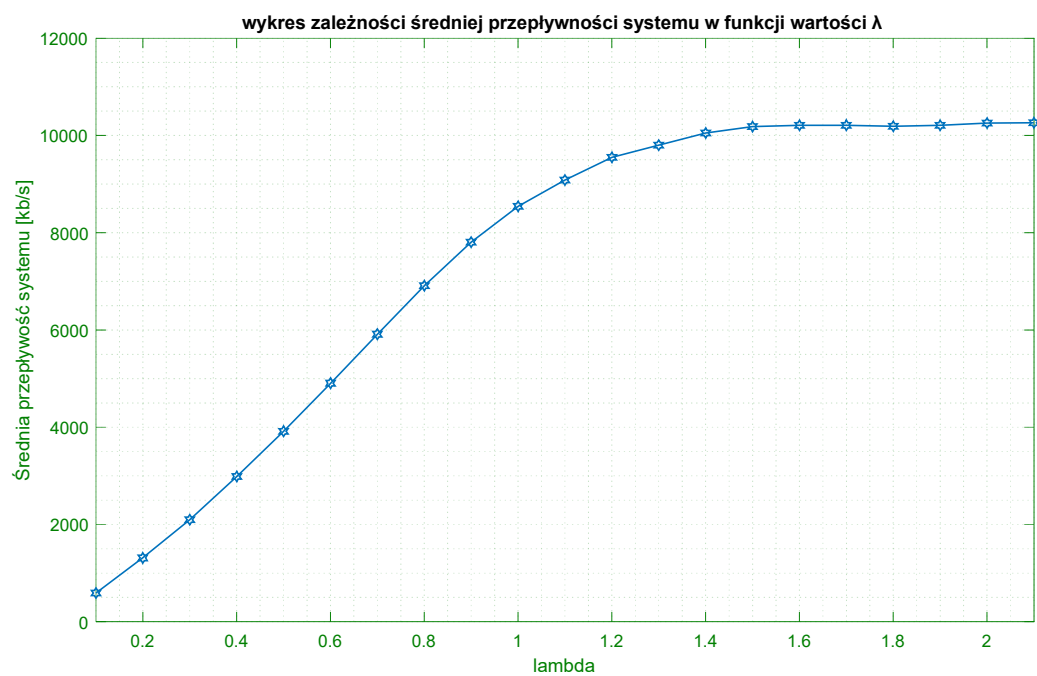
**d. Tabelka z wynikami symulacji dla każdego przebiegu symulacyjnego (dla wszystkich wymaganych parametrów, również z przedziałami ufności) + e) Wyniki końcowe w postaci uśrednionych wyników po wszystkich przebiegach + przedziały ufności dla każdego z trzech.**

lambda: 1.485, czas symulacji: 2000, faza początkowa: 500

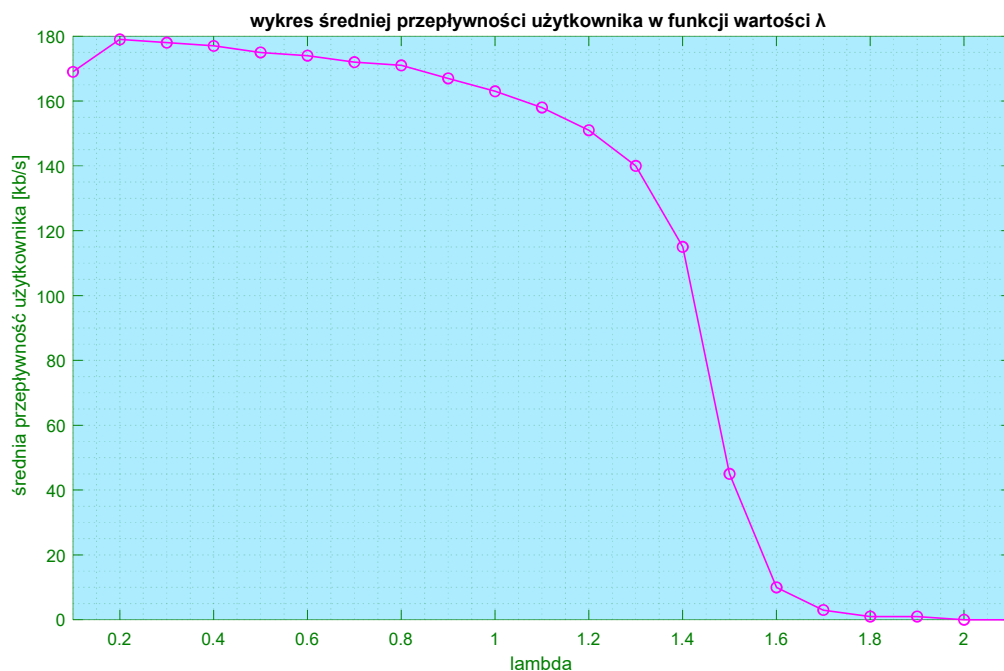
Nr symulacji	Średnia przepływność systemu [kb/s]	Średni czas oczekiwania [ms]	Średnia przepływność użytkownika [kb/s]	Liczba użytkowników
1	9261	104.458	12	5462
2	9189	94.1353	14	5359
3	9144	57.4894	26	5233
4	9091	43.2598	31	5243
5	9137	19.819	79	4912
6	9247	51.2549	29	5378
7	9255	61.2411	22	5255
8	9220	16.6612	84	4932
9	9126	23.3452	68	4986
10	9184	33.8034	52	5041
Średnia	9185.4	50.5467	41.7	5180
-/+ Przedział ufności	[9131.98 , 9238.82]	[23.63 , 77.45]	[17.57 , 65.82]	



**f. wykres wymagane wykresy**



Dodatkowo:



**g. Porównanie wyników z osobą implementującą inny algorytm A (alternatywnie wyniki dla np. innej liczby bloków zasobów)**

	Mój symulator	Symulator kolegi
Metoda	M2	M4
Algorytm	A1b	A2a
Liczba bloków zasobów	25	75
Maks. liczba bloków dla użytkownika	3	5
Czas pomiędzy przydziałami RB	5ms	1ms
Zestawy ziaren	Takie same	

Nr symulacji (zestaw ziaren)	1	1
Lambda	1.485	1.485
Koniec fazy początkowej	500	500
Koniec symulacji	5000	5000
Uzyskane wyniki		
Średnia przepływność systemu [kb/s]	9256	8659
Średnia przepływność użytkownika [kb/s]	9	0.55
Czas oczekiwania [ms]	138.77	1
Liczba wszystkich użytkowników	15786	17116

Nr symulacji (zestaw ziaren)	2	2
Lambda	1.1	1.1
Koniec fazy początkowej	500	500
Koniec symulacji	5000	5000
Uzyskane wyniki		
Średnia przepływność systemu [ $kb/s$ ]	8311	4752.10
Średnia przepływność użytkownika [ $kb/s$ ]	140	0.51
Czas oczekiwania [ $ms$ ]	9.36	1
Liczba wszystkich użytkowników	9307	9981

## 8. Wnioski

Po wielu symulacjach, mogę stwierdzić, iż algorytm Round Robin, nie jest najbardziej optymalnym algorytmem przy dużych wartościach intensywności zgłoszeń. Przy wartości lambdy 1.4 widoczny jest duży skok średniego czasu oczekiwania, ogromny spadek średniej przepływności użytkownika jest bardzo widoczny pomiędzy lambdą 1.4 a 1.5. Średnia przepływność systemu osiąga swój limit przy lambdzie 1.5. Wpływ na tak małą efektywność, oprócz algorytmu miały takie czynniki jak: mała ilość bloków zasobów, a także duży czas przydziału. Intensywność zgłoszeń o wartości 1.485 zapewnia czas oczekiwania nie większy niż 50ms.