



PROJEKTOWANIE SIECI TELEINFORMATYCZNYCH

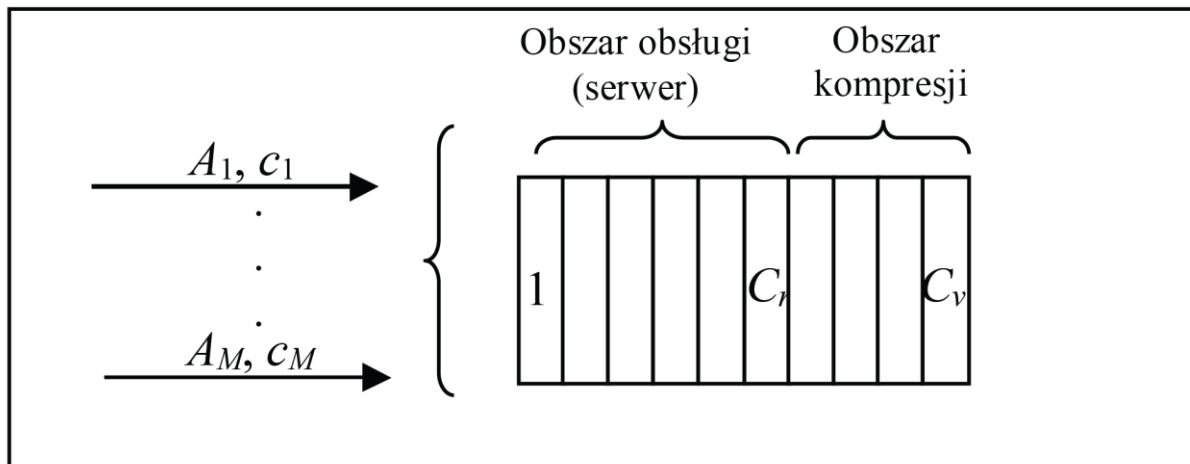
PROJEKT MODELU MATEMATYCZNEGO OKREŚLAJĄCEGO
PRAWDOPODOBIENSTWO BLOKADY W SYSTEMIE Z ZASOBAMI
PEŁNODOSTĘPNYMI Z RUCHEM ELASTYCZNYM I ADAPTACYJNYM (*ZPEA*)

8 lipca 2020

Osoby wykonujące projekt:	Numery indeksu:	Prowadzący:	Data wykonania projektu:	Kierunek studiów:
inż. Karol Nowiński inż. Marcel Adamski inż. Radosław Sęk	121620 131030 131135	prof. dr hab. inż. Maciej Stasiak	6 lipca 2020	Teleinformatyka

1. Wstęp teoretyczny

W ramach projektu zaliczeniowego z przedmiotu Projektowanie Sieci Teleinformatycznych wykonany został program, będący implementacją modelu matematycznego określającego prawdopodobieństwo blokady w systemie z zasobami pełnodostępnymi z ruchem elastycznym i adaptacyjnym. Standardowo ideowy schemat takiego systemu, będącego modelem wielousługowego serwera z kompresją ruchu, przedstawia się w literaturze przedmiotu w sposób następujący:



gdzie:

A_i – średnie natężenie ruchu klasy i (w odniesieniu do JP),

c_i – przepływność strumienia klasy i ,

M – liczba oferowanych strumieni zgłoszeń,

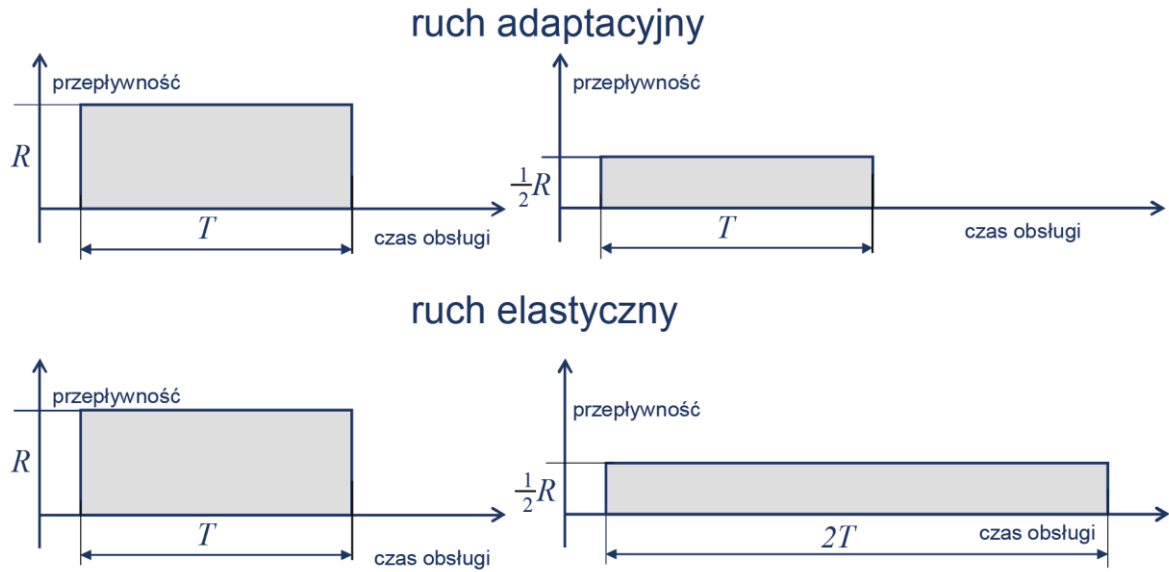
C_r – pojemność rzeczywista,

C_v – pojemność wirtualna.

Zgłoszenia w takim systemie mogą podlegać mechanizmowi kompresji dopóki liczba zajętych JP w zasobach, określana jako suma nieskompresowanych żądań zgłoszeń wszystkich klas, nie przekroczy wirtualnej pojemności C_v . Jeżeli suma żądań obsługiwanych zgłoszeń i nowego zgłoszenia przekroczy pojemność wirtualną C_v , to wówczas nowe zgłoszenie zostanie odrzucone.

Stany n , takie, że $C_r < n \leq C_v$ wyznaczają obszar kompresji ruchu elastycznego/adaptacyjnego. Wybór wartości C_v jest wskaźnikiem tzw. głębokości kompresji, który jest równy stosunkowi wirtualnej pojemności serwera do pojemności rzeczywistej $\frac{C_v}{C_r}$ i określa ile razy może maksymalnie zmniejszyć się całkowita przepływność obsługiwanych zgłoszeń w systemie.

Ideę zarówno ruchu adaptacyjnego, jak i ruchu elastycznego ilustruje poniższy schemat:



W warunkach obciążenia sieci ponad przyjętą wartość pojemności C_r dla ruchu adaptacyjnego następuje zmniejszenie prędkości bitowej (przepływności) bez zmiany czasu emisji. W następstwie takiego rozwiązania część informacji zostaje tracona. Natomiast w przypadku ruchu elastycznego zmniejszeniu przepływności towarzyszy zmiana czasu obsługi. Oba podejścia prowadzą do pogorszenia parametrów jakości obsługi (ang. *Quality of Service*), umożliwiając jednakowoż obsłużenie większej ilości zgłoszeń w systemie.

Rozkład zajętości w projektowanym systemie określony jest na podstawie równań rekurencyjnych:

$$\begin{cases} [P(n)]_{C_v} = \frac{1}{\min(n, C_r)} \left\{ \sum_{j=1}^{M_1} A_j [p(n-t_j)]_{C_v} + \sum_{k=1}^{M_2} A_k \frac{C_r}{n} [p(n-t_k)]_{C_v} \right\} & \text{dla } 0 \leq n \leq C_v, \\ [P(n)]_{C_v} = 0 & \text{dla innych } n, \\ \sum_{n=0}^{C_v} [P(n)]_{C_v} = 1. & \{E_j : j \in N, j \leq M_1\}, \end{cases}$$

Gdzie dla $0 < n \leq C_v$ pierwsza z sum odpowiada za ruch elastyczny, natomiast druga – oznaczona czerwonym okregiem – odpowiada za ruch adaptacyjny.

Prawdopodobieństwo blokady E_i dla każdej klasy zgłoszeń i ($0 < i \leq M$) wyrażone jest poniższą sumą:

$$E_i = \sum_{n=C_v-C_i+1}^{C_v} [P_n]_{C_v}$$

2. Implementacja programu

2.1. Implementacja modelu Kaufmana-Robertsza

W celu wykonania poniższego projektu systemu z zasobami pełnodostępnymi z ruchem elastycznym i adaptacyjnym zaimplementowano wstępnie model Kaufmana-Robertsza, na którym to oparto dalsze rozwijanie programu.

2.1.1. Wyniki działania programu

W ramach testowania programu i analizy otrzymanych w wyniku działania programu wartości wynikowych posłużono się wartościami referencyjnymi dostarczonymi przez prowadzącego.

V=50				
i	t _i	A _i	E _i	E _i [<i>ref</i>]
1	1	8,33333333	0,0042123186728921	0,00421223
2	5	1,666676667	0,0273542559100583	0,0273537
3	10	0,83333333	0,0753867491840429	0,07538522
V=100				
i	t _i	A _i	E _i	E _i [<i>ref</i>]
1	2	10	0,006314231462771065	0,00631423;
2	7	2,85714286	0,026987473320365444	0,02698746
3	11	1,81818181	0,049456912907628714	0,04945689;
V=1000				
i	t _i	A _i	E _i	E _i [<i>ref</i>]
1	2	80	0,0009797790677551423	0,00097978
2	5	32	0,0025238350216688953	0,00252384
3	8	20	0,004160927789625112	0,00416093
4	15	10,66666667	0,008367226773222968	0,00836723
5	25	6,40000000	0,01541068926192425	0,01541069

2.1.2. Kod programu dla modelu Kaufmana-Robertsza

```
def Kaufman_Roberts(V,M,A,t):
    for i in range(0,V+1):
        P.append(1)
    #wzór Kaufmana-Robertsza
    for n in range(1,V+1):
        sum=0
        for i in range(0,M+1):
            if n >= t[i]:
                sum+=A[i]*t[i]*P[n-t[i]]
        P[n]=sum/n

P=[]
#user input
#pojemność systemu
V=int(input("Podaj pojemność systemu [V]:"))
#ilość klas usług
M=int(input("Podaj ilość klas usług (1-5) [M]:"))

#natężenie ruchu klasy
A=[0]
for i in range(1,M+1):
    A.append(float(input("Podaj natężenie ruchu klasy A[%s]:"%i)))
A.append(0)

#ilość potrzebnych zasobów
t=[0]
for i in range(1,M+1):
    t.append(int(input("Podaj ilość potrzebnych zasobów t[%s]:"%i)))
t.append(0)
print()

#obliczenie prawdopodobieństw
Kaufman_Roberts(V,M,A,t)

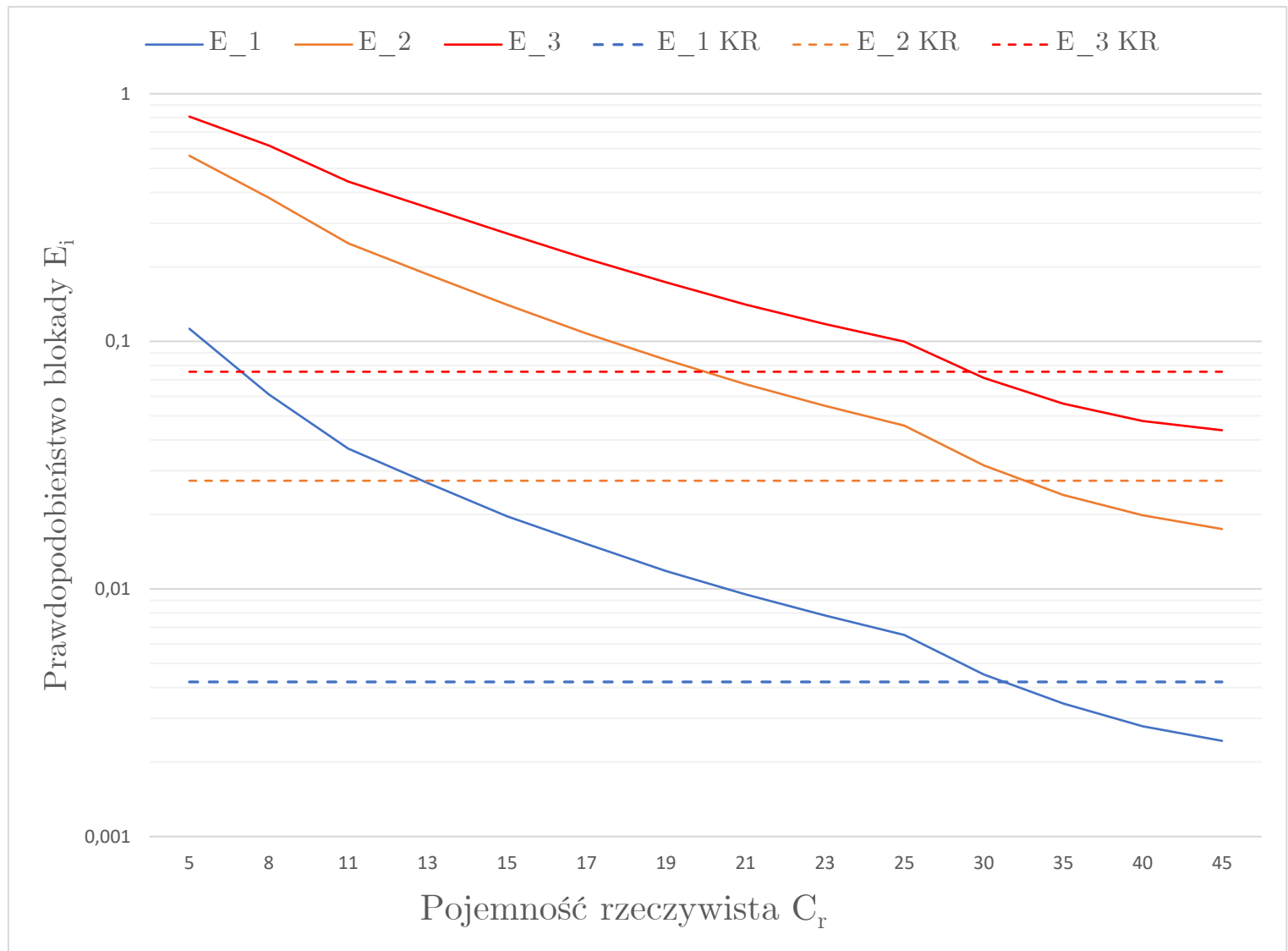
#normalizacja
total=0
for i in range(1,V+1):
    total+=P[i]
for i in range(1,V+1):
    P[i]/=total

#prawdopodobieństwa blokady
E=[0]
for i in range(1,M+1):
    E.append(0)
    for n in range(V-t[i]+1,V+1):
        E[i]+=P[n]
    print("E(%s)=%s"%(i,E[i]))
```

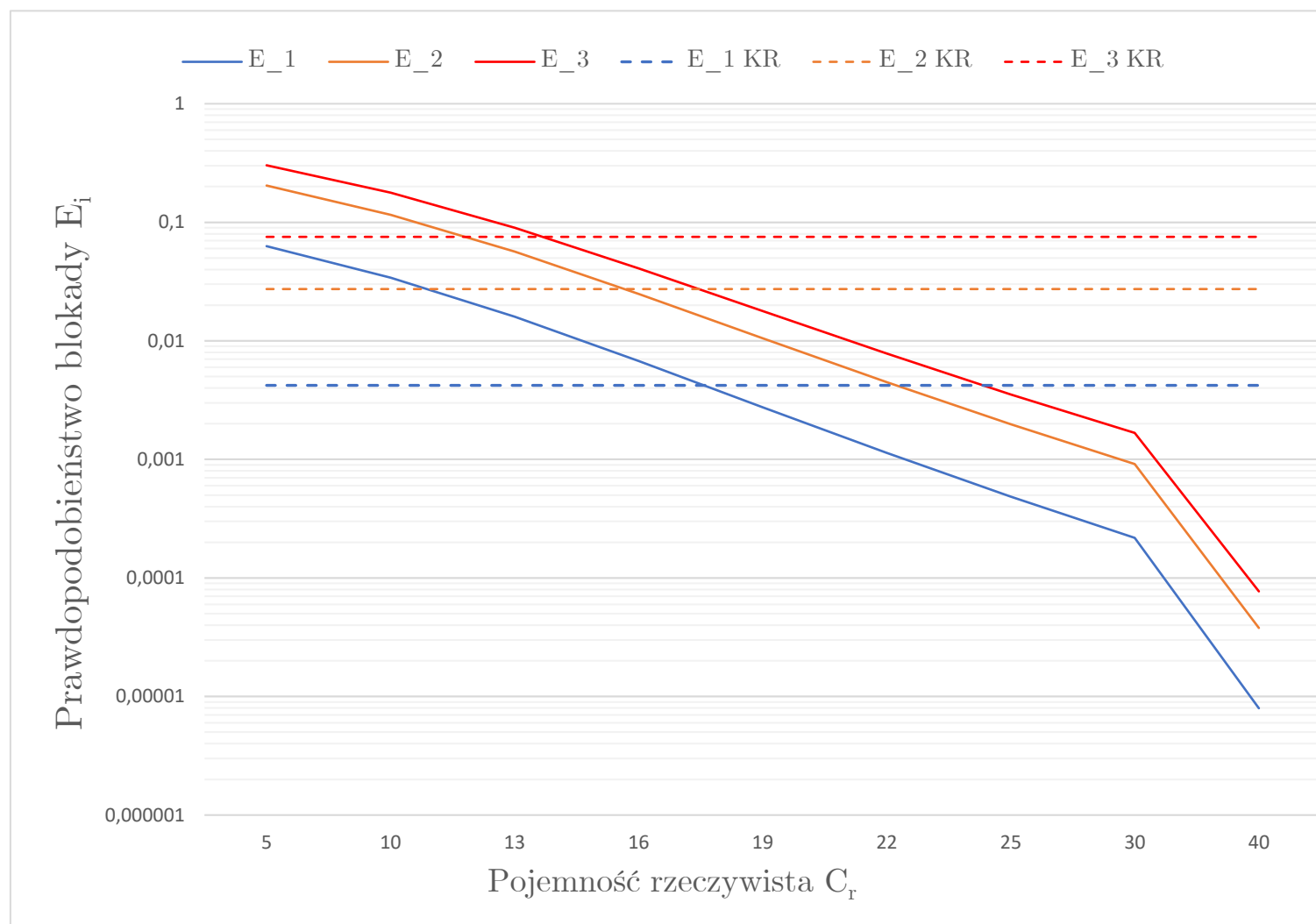
2.2. Model systemu z zasobami pełnodostępnymi z ruchem elastycznym

2.2.1. Wyniki działania programu

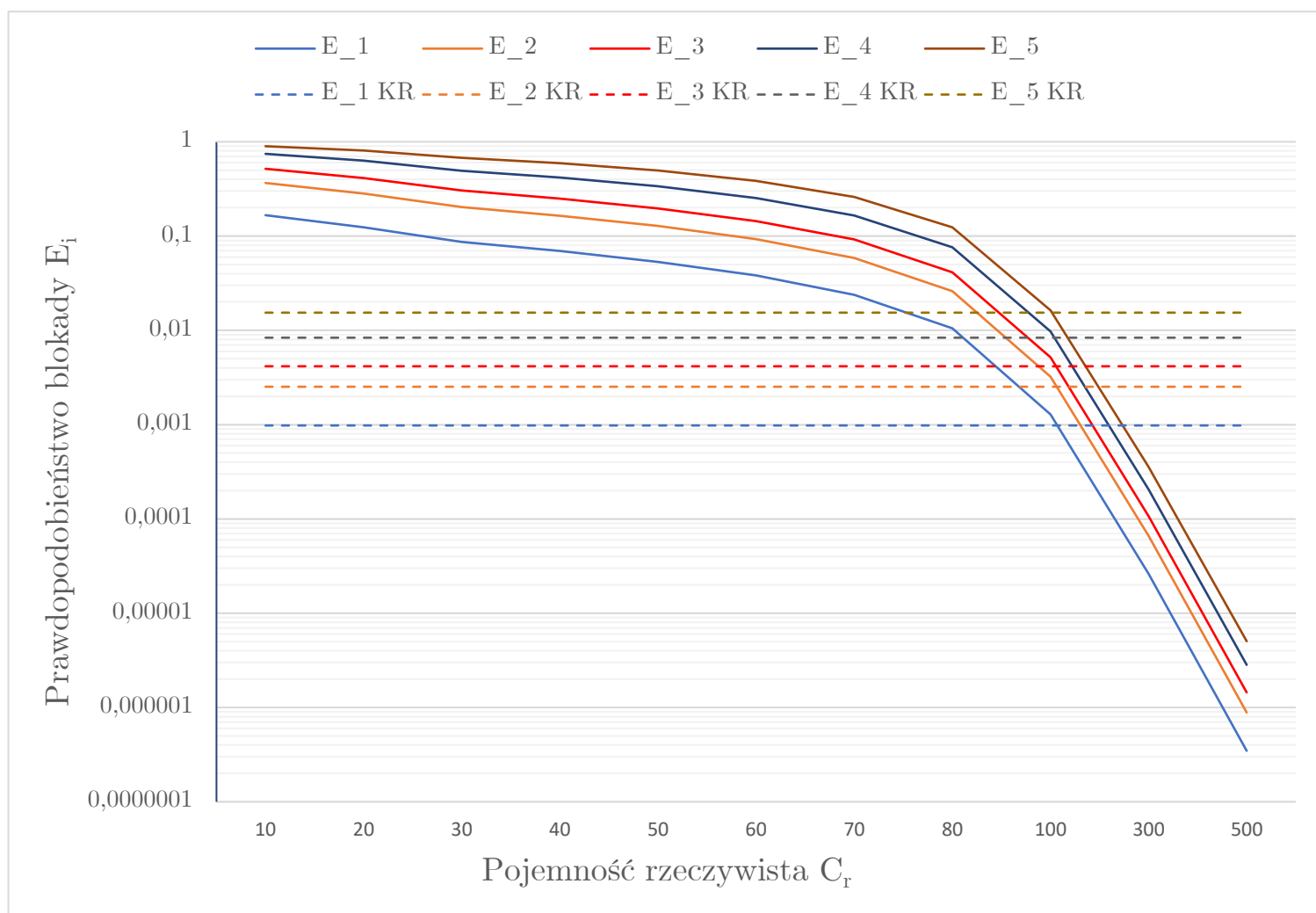
- dla $C=50$



• dla $C=100$



• dla $C=1000$



2.2.2. Kod programu

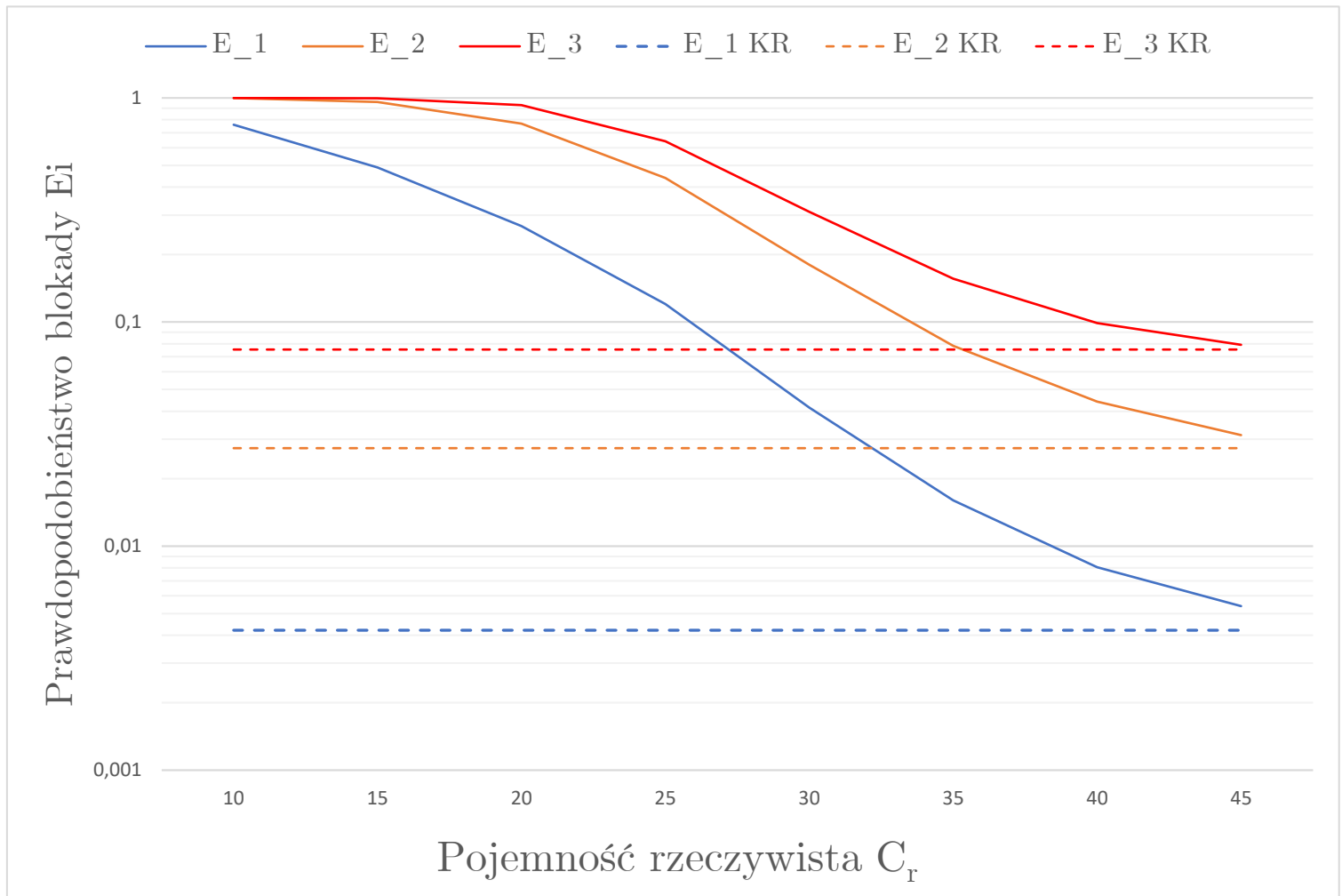
```
def ZP_elastic(V,M,A,t,cr):
    for i in range(0,V+1):
        P.append(1)
    #wzór na ruch elastyczny
    for n in range(1,V+1):
        sum=0
        for i in range(0,M+1):
            if n >= t[i]:
                sum+=A[i]+t[i]*P[n-t[i]]
        div=min(n,cr)
        P[n]=sum/div

P=[]
#user input
#pojemnosc rzeczywista
cr=int(input("Podaj pojemność rzeczywistą [cr]:"))
#pojemność wirtualna
cv=int(input("Podaj pojemność wirtualną [cv]:"))
V=cv
#ilość klas usług
M=int(input("Podaj ilość klas usług (1-5) [M]:"))
#natężenie ruchu klasy
A=[0]
for i in range(1,M+1):
    A.append(float(input("Podaj natężenie ruchu klasy A[%s]:"%i)))
A.append(0)
#ilość potrzebnych zasobów
t=[0]
for i in range(1,M+1):
    t.append(int(input("Podaj ilość potrzebnych zasobów t[%s]:"%i)))
t.append(0)
#obliczenie prawdopodobieństw
ZP_elastic(V,M,A,t,cr)
#normalizacja
total=0
for i in range(1,V+1):
    total+=P[i]
for i in range(1,V+1):
    P[i]/=total
print()
#prawdopodobieństwa blokady ruchu elastycznego
E=[0]
for i in range(1,M+1):
    E.append(0)
    for n in range(V-t[i]+1,V+1):
        E[i]+=P[n]
    print('E({})={:.20f}'.format(i,E[i]))
```

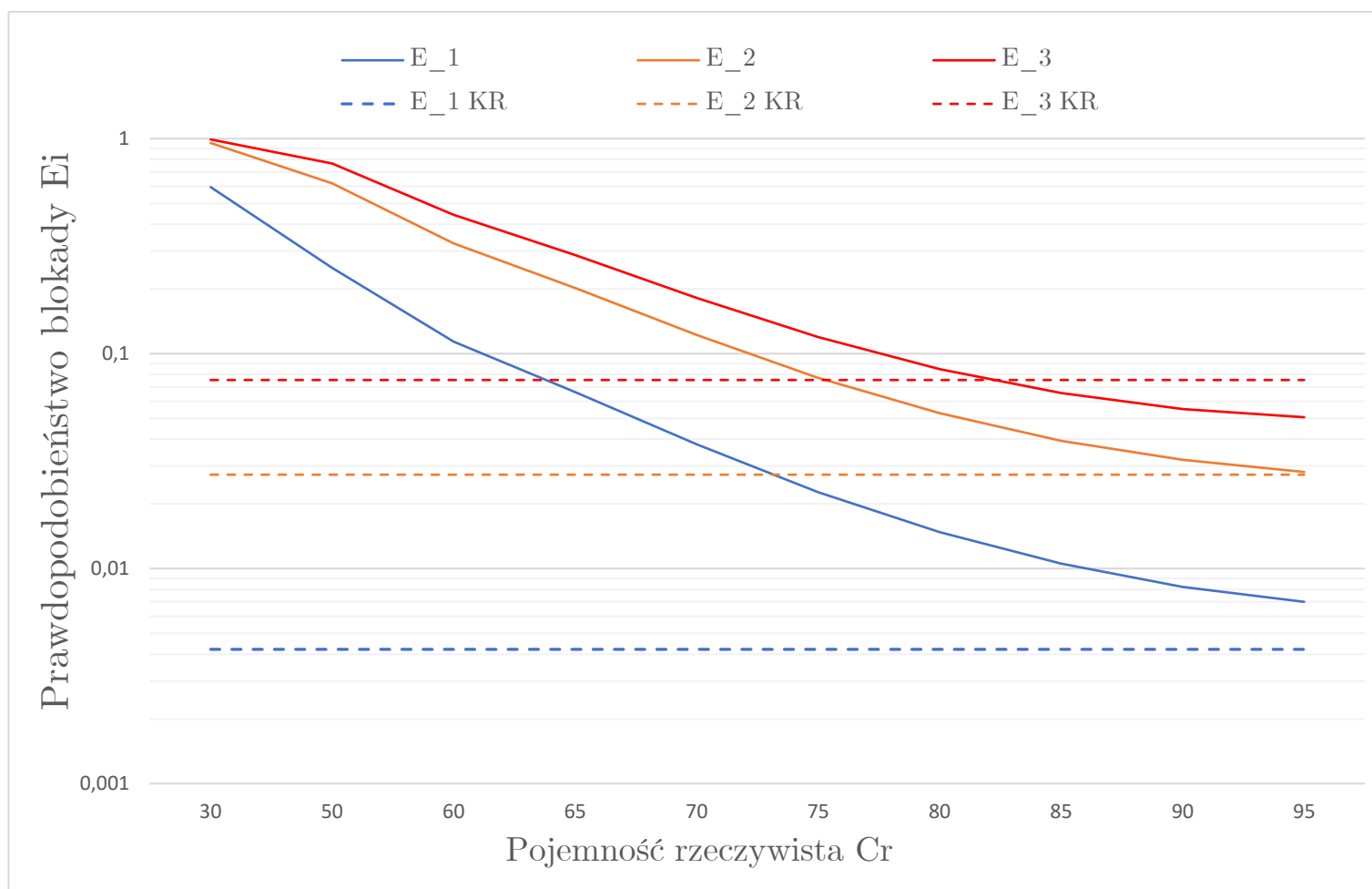
2.3. Model systemu z zasobami pełnodostępnymi z ruchem adaptacyjnym

2.3.1. Wyniki działania programu

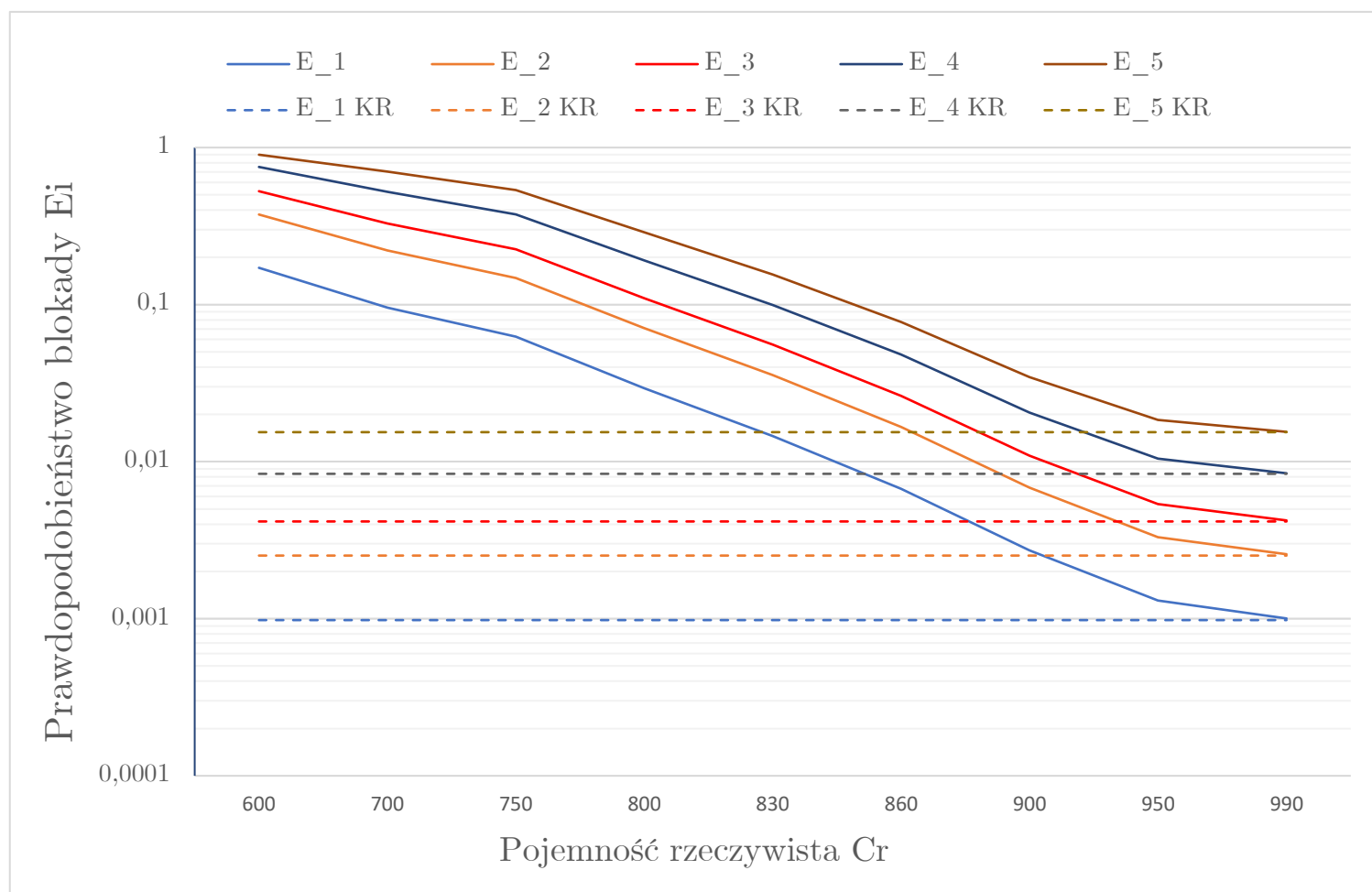
- dla $C=50$



• dla $C=100$



• dla $C=1000$



2.3.2. Kod programu

```
def ZP_adaptive(V,M,A,t,cr):
    for i in range(0,V+1):
        P.append(1)
    #wzór na ruch adaptacyjny
    for n in range(1,V+1):
        sum=0
        for i in range(0,M+1):
            if n >= t[i]:
                if n <= cr:
                    sum+=A[i]*t[i]*P[n-t[i]]
                else:
                    sum+=A[i]*t[i]*(n/cr)*P[n-t[i]]
        div=min(n,cr)
        P[n]=sum/div

P=[]
#user input
#pojemnosc rzeczywista
cr=int(input("Podaj pojemność rzeczywistą [cr]:"))
#pojemność wirtualna
cv=int(input("Podaj pojemność wirtualną [cv]:"))
V=cv
#ilość klas usług
M=int(input("Podaj ilość klas usług (1-5) [M]:"))
#natężenie ruchu klasy
A=[0]
for i in range(1,M+1):
    A.append(float(input("Podaj natężenie ruchu klasy A[%s]:"%i)))
A.append(0)
#ilość potrzebnych zasobów
t=[0]
for i in range(1,M+1):
    t.append(int(input("Podaj ilość potrzebnych zasobów t[%s]:"%i)))
t.append(0)
#obliczenie prawdopodobieństw
ZP_adaptive(V,M,A,t,cr)
#normalizacja
total=0
for i in range(1,V+1):
    total+=P[i]
for i in range(1,V+1):
    P[i]/=total
print()
#prawdopodobieństwa blokady ruchu adaptacyjnego
E=[0]
for i in range(1,M+1):
    E.append(0)
    for n in range(V-t[i]+1,V+1):
        E[i]+=P[n]
    print('E({})={:.20f}'.format(i,E[i]))
```

3. Wnioski

W ramach projektu zaliczeniowego z przedmiotu Projektowanie Sieci Teleinformatycznych wykonany został program, będący implementacją modelu matematycznego określającego prawdopodobieństwo blokady w systemie z zasobami pełnodostępnymi z ruchem elastycznym i adaptacyjnym.

Projekt w swojej ostatecznej postaci bazuje na modelu Kaufmana-Robertsza, który to został rozbudowany w ramach opracowywania danych modeli – kolejno dla ruchu elastycznego oraz adaptacyjnego. Wynikiem są dwa osobne podprogramy realizujące modele systemu dla obydwu tych ruchów.

Autorzy projektu przyjęli koncepcję traktowania wartości pojemności realnej C_r jako wartości zmiennej dla wykreślania wykresów prawdopodobieństwa blokady dla poszczególnych typów ruchu. Wyniki działania danych podprogramów zostały zatem przedstawione na wykresach, jako zależność wartości prawdopodobieństwa blokady od wartości pojemności realnej C_r , przy założeniu stałej wartości C (V dla modelu Kaufmana-Robertsza). Na każdym z wykresów znajduje się również referencyjna wartość prawdopodobieństwa blokady dla modelu Kaufmana-Robertsza, określona dla tej samej wartości pojemności (tj. pojemności całkowitej, na którą w modelach z ruchem elastycznym/adaptacyjnym składa się pojemność realna oraz wirtualna).

Sprawozdanie z przebiegu wykonywania projektu zostało w głównej jego części podzielone na kilka osobnych sekcji – autorzy najpierw przybliżyli kod programu zaimplementowanego modelu Kaufmana-Robertsza oraz wyniki jego działania porównane z referencyjnymi wartościami otrzymanymi od prowadzącego, a następnie przedstawili jego rozbudowanie dla modeli z ruchem elastycznym oraz adaptacyjnym wraz z analizą wyniku działania tych poszczególnych podprogramów.

W wyniku analizy wykresów zależności wartości prawdopodobieństwa blokady od wartości pojemności realnej C_r , zauważono, iż dla modelu systemu z ruchem elastycznym zwiększanie parametru C_r prowadzi do zmniejszenia prawdopodobieństwa blokady dla wszystkich klas zgłoszeń.

Również w przypadku ruchu adaptacyjnego zwiększenie wspomnianego parametru daje podobny rezultat.

Natomiast porównując wyniki poszczególnych ruchów z referencyjną wartością prawdopodobieństwa blokady wyliczoną dla modelu Kaufmana-Robertsza, można zauważyć, że o ile model systemu z ruchem elastycznym prowadzi do lepszych rezultatów w stosunku do referencji, o tyle model systemu z ruchem adaptacyjnym zbliża się wartością prawdopodobieństwa blokady do wartości referencyjnej.

Autorzy projektu zakładają, że może to być spowodowane naturą obydwu tych ruchów. W warunkach obciążenia sieci ponad przyjętą wartość pojemności C_r dla ruchu adaptacyjnego następuje zmniejszenie prędkości bitowej (przepływności) bez zmiany czasu obsługi. Natomiast w przypadku ruchu elastycznego zmniejszeniu przepływności towarzyszy również zmiana czasu obsługi. Próba zachowania pierwotnie przesyłanej informacji może w konsekwencji prowadzić do wyższego prawdopodobieństwa blokady, spowodowanej wspomnianym wyżej wydłużeniem czasu obsługi.

4. Bibliografia

M. Stasiak, *Queuing systems for the internet*, IEICE Transactions on Communications, vol. E99-B, no. 6, 1224-1242, jun 2016