ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | Е.А.Бакин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
|  |
| Дисциплина: НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА |

РАБОТУ ВЫПОЛНИЛ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4913М |  |  |  | В.К.Михайлов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2020

**Цель работы:** Приобретение практических навыков по изучению среднего времени ожидания абонента на базовой станции и вероятности возникновения коллизий.

**Вариант:**

**Среднее время ожидания**

N=20 – количество абонентов

p – вероятность прихода пакета

FN = 200 – количество фреймов

Рассчитать среднее время ожидания пакета в зависимости от нагрузки станции.

**Вероятность возникновения коллизии**

N=30 – количество абонентов

p – вероятность прихода пакета

Рассчитать вероятность повреждения пакетов в зависимости от нагрузки станции.

**Листинг:**

Листинг 1 - Система с фреймами

|  |
| --- |
| # -\*- coding: utf-8 -\*-  # LR 1  import random  import numpy as np  import matplotlib  import matplotlib.pyplot as plt  N = 20 # Number of abonents  p = 0.01 # p of message  FN = 200 # number of frames  Slots = np.zeros(N) # slots in frame  Buf = np.zeros(N) # message buffer  FrameTimer = [[] \* 1 for i in range(N)]  SrT = 0 # average time  MsgIn = 0  ServNum = 0 # messages served  BufNum = 0 # messages buffered  def reset():  global Slots, Buf, FrameTimer, SrT, ServNum, BufNum, MsgIn  Slots = np.zeros(N) # slots in frame  Buf = np.zeros(N) # message buffer  FrameTimer = [[] \* 1 for i in range(N)]  SrT = 0 # average time  MsgIn = 0  ServNum = 0 # messages served  BufNum = 0 # messages buffered  def gen(p):  frame = np.random.binomial(1, p, (N, N)).tolist()  return frame  def TakeFromBuffer():  slots = np.zeros(N)  for i in range(N):  if Buf[i] > 0:  Buf[i] -= 1  slots[i] = 1  global SrT  SrT+= FrameTimer[i][0]  FrameTimer[i].pop(0)  return slots  def AddTime(timer):  for i in range(len(timer)):  for j in range(len(timer[i])):  timer[i][j]+=1  return timer  def Simulate():  global FrameTimer,ServNum,BufNum,Buf,p,MsgIn,SrT  for i in range(FN):  FrameTimer = AddTime(FrameTimer)  frame = gen(p)  MsgIn +=sum(sum(frame,[]))  # print(frame)  # print("--- Frame ", i,"---")  # print("Buffer before", Buf)  Slots = np.zeros(N)  Slots = TakeFromBuffer()  # print("Slots", Slots)  # print("Buffer after", Buf)  # print(frame)  for j in range(N):  if Slots[j] == 0 and frame[j][j] == 1:  frame[j][j] = 0  # print(j,frame[j])  Slots[j]=1  ServNum += 1  BufNum += sum(frame[j])  ServNum += sum(Slots)  for k in range(N):  if frame[j][k] >0:  FrameTimer[k].append(1)  Buf = np.add(Buf, frame[j])  SrT+= sum(sum(FrameTimer,[]))  out = SrT/MsgIn  #print(out)  return out  SrtList =[]  px=[]  for i in range(9):  p+=(1/N)/9  print(i,'/10')  px.append(p)  reset()  SrtList.append(Simulate())  print(p)  print(SrtList)  plt.plot(px,SrtList)  plt.show() |

Листинг 2 - Система без фреймов

|  |
| --- |
| # -\*- coding: utf-8 -\*-  # LR 1 no frames  import random  import numpy as np  import matplotlib  import matplotlib.pyplot as plt  N = 30 # Number of abonents  p = 0.001 # p of message  collisions = 0  passed = 0  pColl = []  px=[]  def gen(p):  frame = np.random.binomial(1, p, N).tolist()  return frame  def reset():  global collisions, passed  collisions, passed = 0, 0  def Simulation():  simLen = 500 # sinulation lenght  reset()  tempCollisions =0  global collisions, passed, pColl  for i in range(simLen):  if (sum(gen(p))) > 1:  collisions += 1  pColl.append(collisions / simLen)  for i in range(190):  p+=1/200  px.append(p)  Simulation()  if pColl[-1] == 1.0: break  plt.plot(px,pColl)  plt.show()  #Simulation()  #print(pColl)  #print(gen(1)) |

**Результаты работы программ**

Результаты выполнения для системы с фреймами. Для 10 абонентов и 200 фреймов.

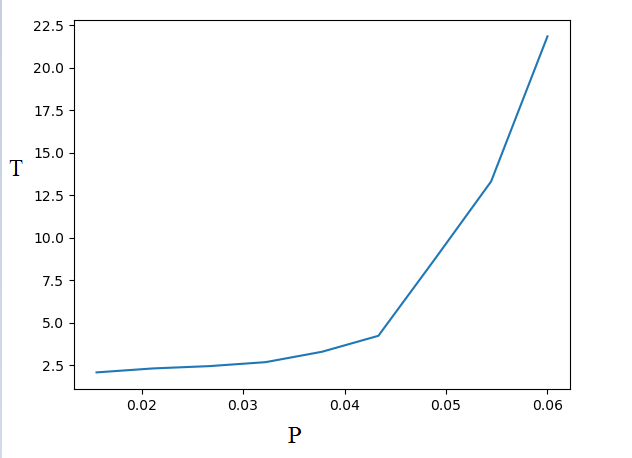
****

Рисунок 1 – Зависимость времени ожидания в очереди от вероятности прихода пакета

Результаты выполнения для системы без фреймов. Для 30 абонентов.

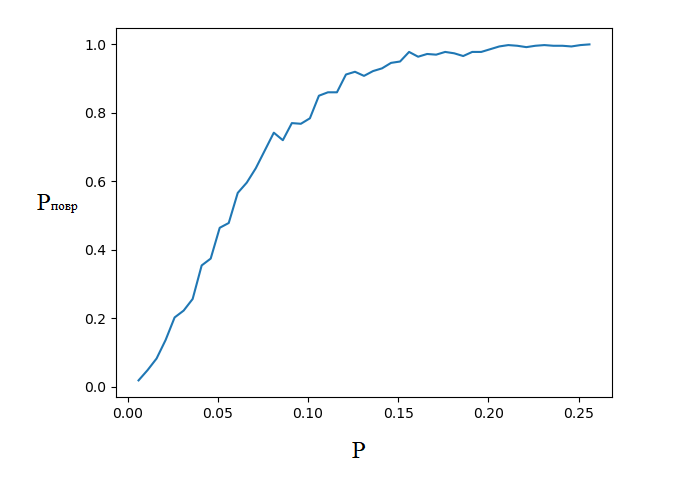


Рисунок 2 - Зависимость вероятности коллизии от вероятности прихода пакета

**Вывод**

В ходе выполнения лабораторной работы было оценено среднее время ожидания пакета для станции с фреймами и вероятность коллизий для станции без фреймов.

Для станции с фреймами, чем больше вероятность прихода пакета, тем дольше абоненты ожидают в очереди.

Для станции без фреймов, чем больше вероятность прихода пакета, тем больше становится вероятность коллизии.