CS 350 Homework 02 Tim Duffy

1) a. What is your guess as to the availability of the service when offered using the new 2-server configuration? Show all calculations.

If the servers are identical, and both have 95% availability, the first server will be down 5% of the time and the second server will take over (still with availability of 95%). This will give the overall system an availability of 0.95 + (0.95*0.05) = 0.95 + 0.0475 = 0.9975 = 99.75%

b. Suggest a pattern of server failures that will end-up in an "over estimation" (or an "under estimation") of the service availability using your friend's measurement approach?

If we test the server every minute, on the minute, we may end up with an overestimation of our servers availability. If the servers are up every time we test the system, but they are down for the majority of time between tests, we will overestimate the availability.

c. Suggest a better measurement approach.

Instead, the pings to test the system should be sent in completely random intervals. This will sample the availability of the system in such a way that we get a better approximation of the availability of the service.

d. What assumption needs to hold for the two-server configuration suggested by your friend to yield the improvement in availability that your friend had hoped for?

In order for the expected improvement to actually occur, the servers both need to have an availability of 0.95, and the second server needs to successfully service 95% of requests that fail on the first server.

2) a. What is the probability distribution for the random variable Tn? Write down the PDF of Tn, when n=100.

$f(x) = 1/(\sigma * \sqrt{2\pi}) * e^{-0.5 * (100 - 2)\text{^}2 / \sigma\text{^}2}$

b. What is the mean and standard deviation of Tn, when n=100?

$\mu = 100 * 2 =$ **200 msecs**
$\sigma = \perp\sqrt{100} * 0.8$ msec $= 10 * 0.8 =$ **8 msecs**

c. What is the probability that the retrieval time for a file that consists of 100 blocks will be between 185 and 190 msec?

z(185) = (185 - 200) / 8 = -15/8 = - 1.875
P(185) = 1 - 0.9696 = 0.0304
z(190) = (190 - 200) / 8 = -10/8 = -1.25
P(190) = 1 - .8944 = 0.1056
P(190) - P(185) = 0.1056 - 0.0304 = **0.0752 = 7.52%**

3) a. Write down the formula for the probability distribution of the number of packets arriving to the adapter in one millisecond.

$\lambda = 100$ packets/sec

$f(x) = ((\lambda t)^x/x!) * e^{-\lambda t}$ where x is 0,1,2...
$f(x) = ((100*0.001)^x/x!) * e^{-(100*0.001)}$ where x is 0,1,2...
$f(x) = ((0.1)^x/x!) * e^{-(0.1)}$ where x is 0,1,2...

b. What is the probability that more than 1 packet will be received within a 10-millisecond period of time?

P(>1 arrival in a 10 msec period) = 1 - P(1 arrival in 10 msec) - P(0 arrivals in 10 msec)

$f(1) = ((100 * 0.01)^1/1!) * e^{-(100 *0.01)} = e^{-1} = 0.367879441$
$f(0) = ((100 * 0.01)^0/0!) * e^{-(100 *0.01)} = e^{-1} = 0.367879441$

1 - f(1) - f(0) = 1 - 0.367879441 - 0.367879441 = 0.264241118 **= 26.42%**

c. Write down the formula for the probability distribution of the inter-arrival time of packets.

Only 1 of the following events can occur:
- no packets arrive in interval *t*
- at least one packet arrives in interval *t*

So, 1 = P(no arrivals in *t*) + P(>=1 arrival in *t*)
$1 = F(t) + e^{-\lambda t}$ => F(x) = P(no arrivals in interval *t*)
$\mathbf{F(t) = 1 - e^{-\lambda t}}$

d. What is the probability that two packets will be separated by more than 20 milliseconds?

P(arrivals are separated by 20 msec or more) = $F(20) = 1 - e^{-\lambda t}$
$= 1 - e^{-(100 * 0.02)} = 1 = 1 - e^{-2} = 1 - 0.135335283 =$ **0.864664717 = 86.47%**

e. What is the standard deviation of the inter-arrival time of packets?

Because this is an exponential distribution, both the mean and the standard deviation are equal to $1/\lambda = 1/100 = 0.01$ seconds = **10 milliseconds**

f. What is the probability that the time between the arrival of a packet and the arrival of the 100th packet following that first packet is less than 1.1 seconds?

Given that 1 packet already arrives at time t=0, we can use the CLT to compute the probability that the 100th packet arrives less than 1.1 seconds later.

$\mu = 100 * 1/\lambda = 1000$ milliseconds = 1 second
$\sigma = \sqrt{100} * 1/\lambda = 100$ milliseconds = 0.1 second

$f(x) = 1 / (0.1*\sqrt{2\pi}) * e^{-0.5 * [(1.1 - 1)^2 / (0.1)^2]} = (1/0.250662827) * e^{-0.5} =$ **2.4197 %**

4) Given: λ = 100 packets/sec = 0.1 packets/msec T_s = 8 msec

g. What is the capacity of the Ethernet adapter?

$\mu = 1/T_s = 0.125$ msec/packet => $\rho = \lambda/\mu = 0.1/0.125 =$ **0.8 packets/msec**

h. Write down the probability distribution of the number of packets at the Ethernet adapter.

$f(x) = (\lambda)^x/x!) * e^{-\lambda}$
$f(x) = (0.1)^x/x!) * e^{-0.1}$

i. How many packets do you expect to find in the buffer of the adapter at any point in time (on average)?

Using Little's formula, we get $T_w = \rho/\mu(1-\rho)$ => w(# of packets in buffer) = $T_w * \lambda$
So, the average number of packets in the buffer (q) is :
$\lambda * \rho/\mu(1-\rho) = 0.1 * (0.8/0.125)*(1-0.125) = 0.1*6.4*0.875 =$ **0.56**

Since the average service time for each packet is faster than the flow of incoming packets, there is an average of less than 1 packet in the buffer at any time.

j. What is the average waiting time in the buffer (i.e. how long is a packet buffered until the ISR starts processing it)?

$T_q = \lambda * q = 0.1$ packets/msec * 0.56 packets **= 0.056 msecs**

k. What is the slowdown caused by queuing at the Ethernet adapter?

Slowdown = T_q/T_s = (0.056 / 8) msec = **0.007 msec**

l. The web server is under a Denial of Service (DoS) attack, causing the system to crash. What is the minimum rate of adversarial traffic that must have been injected to cause the crash?

Assume that a DOS occurs when the system reaches 90% of its capacity.

0.9 * 0.8 packets/msec = 0.72 packets/msec = 720 packets/second

There is already 100 packets/sec moving through the system. So an adversary must inject an additional 620 packets/sec into the network.

5)

```
import java.util.Random;
import static java.lang.Math.*;
import java.io.*;
import java.util.Scanner;

class hw02_Q5_Tim_Duffy {
  public static void main(String[] args) {

    Scanner console = new Scanner(System.in);

    System.out.println("Input a vaule for U (between 0 and 1) for the mean of the normal
distribution:");
    double U = console.nextDouble();
    console.nextLine();

    System.out.println("Input a vaule for S, the standard deviation of the normal distribution:");
    double S = console.nextDouble();
    console.nextLine();

    double sum = 0;
    for (int i= 0; i<=30; i++){
      sum += Grand(U, S);
    }

    System.out.println("The mean of these 30 samples is " + (sum/30));
  }

  public static double Zrand(){
    /* Random number generator - This function returns a vaule
     * between 0.0 to 1.0 with a uniform distribution */
    Random rndm = new Random();
    return rndm.nextDouble();
  }

  public static double Grand(double U, double S){

    double z = Zrand();
    double x = ((z*S) + U);

    return x;
  }
}
```