Wesleyan University, Spring 2023, COMP 332

Homework 1: Metrics and tic-tac-toe

Due by 11:59pm on February 7, 2023

---

1. Written problems (10 points)

Problem 1. *Suppose two hosts, A and B, are connected by a 10 Mbps link. The length of a packet is 2 Kb (Kilobits). The length of the link is 50 km. Assume that signals propagate at the speed of light.*

a: *What is the propagation delay from A to B? Recall that propagation delay is the amount of time from when the first bit of the packet is transmitted at A, until it is received at B.*

b: *What is the transmission delay of the packet at A? Recall that transmission delay is the time from when the first bit of the packet is sent into the wire to when the last bit is sent into the wire.*

c: *How long must a packet be so that B receives the first bit at the same time that A sends the last bit?*

Problem 2. *Suppose a group of users share a 1 Mbps link. Each user requires 100 Kbps when transmitting, but each user transmits only 10% of the time.*

a: *When circuit switching is used, how many users can be supported?*

b: *Suppose packet-switching is used. What is the probability that a given user is transmitting? Now let's examine how to compute the probability that exactly k of n users are transmitting. Recall that for n independent experiments, each of which succeeds with probability p, the binomial distribution gives the distribution over the number of experiments that succeed. For instance, the probability that exactly k experiments succeed (e.g., k coin flips come up heads or k users transmit simultaneously) is given by:*

$$P(k) \quad = \quad \binom{n}{k} p^k (1-p)^{n-k}$$

Suppose there are 40 users of the network. Use the binomial distribution to compute the probability that more than 10 of the 40 users are transmitting simultaneously at any given time. Note: You may write a program to compute this probability.

c: *Suppose you must provision a packet-switching network. Explain why it is important to have accurate estimates of the number of users, the probability with which they transmit, and the expected bandwidth they use when they do transmit.*

```
> python3 tictactoe.py
==================
| TicTacToe Game |
==================
Enter number of rows in TicTacToe board: 3
_  _  _
_  _  _
_  _  _
Choose row [0-2]: 1
Choose col [0-2]: 2

User move:
_  _  _
_  _  o
_  _  _

Server move:
x  _  _
_  _  o
_  _  _

Choose row [0-2]: 3
Enter row: 2
Choose col [0-2]: 2

User move:
x  _  _
_  _  o
_  _  o

Server move:
x  _  x
_  _  o
_  _  o

Choose row [0-2]: 2
Choose col [0-2]: 0

User move:
x  _  x
_  _  o
o  _  o

Server move:
x  _  x
_  x  o
o  _  o

Choose row [0-2]: 2
Choose col [0-2]: 1

User move:
x  _  x
_  x  o
o  o  o

Winner:  o !
```

FIGURE 1. A sample session of the tic-tac-toe game.

PROBLEM 3. *Download and read the Request For Comments (RFC) titled "RFC 3271: The Internet is for Everyone" found at:* `https://tools.ietf.org/html/rfc3271` *This RFC was written in 2002. How well do you think the RFC predicted the future state of the internet, now 16 years later?*

2. CODING AND HANDS-ON PROBLEMS (10 POINTS)

PROBLEM 4. *The goal of this question is to make sure you have Python3 installed on your computer, (re)familiarize you with Python, and start building the components necessary for a network-based tic-tac-toe game. In this question, you will implement a non-network version of tic-tac-toe; one of the users will be a random strategy that you will implement, and the other user will be you.*

Part 1. *Make sure you have Python3 installed on your machine. You can find Python pointers for installation, tutorials etc., on the course webpage:*

`http://vumanfredi.wescreates.wesleyan.edu/teaching/comp411-f22/resources.html`

Part 2. *Your goal is to implement a version of tic-tac-toe that has a user interaction like what is shown in Figure 1. If you are not familiar with tic-tac-toe, you can find more information here:*

`https://en.wikipedia.org/wiki/Tic-tac-toe`. *Some starter code has been provided for you in* `tictactoe.py`, *which contains three classes:*

(1) `Board`. *This class creates and keeps track of the state of the TicTacToe board. You may want to add additional methods to display the board on the screen, check for a winner, and so on. If someone wants to implement a graphic visualizer of the state of the of the game, I will share it with the class.*

(2) `TicTacToe`. *This class keeps track of the state a TicTacToe game, with methods to be called when the server moves, when the user moves, and so on. For the Server's tic-tac-toe strategy, you may choose a random strategy, or you may choose to implement something more intelligent. For the user's strategy you should query the user.*

(3) `Server`. *This class will create a new tic-tac-toe game in its* `play` *method, and then enter a while loop, alternating between querying the user and the server for moves, as long as the game has not finished.*

You should ask the user for the number of rows (n) in the board to allow an n x n game board rather than just a 3x3 board. The winner of the game must then have all n diagonal elements set.

Part of my goal with this question is to give you some flexibility (and creativity) in architecting your TicTacToe code, so I have not included stubs for all of the methods you might want and you are free to change argument lists for the methods given. You also have some flexibility in creating additional classes to use, or may choose not to use the `Board` *class. You must however use both the* `TicTacToe` *and* `Server` *classes.*

3. Submission

Submit your written work as `hw1.pdf`, and your code as `hw1.py` to the Google Drive directory I have created for you named `comp332-s23-USERNAME/hw1/`. You should replace `USERNAME` with your Wesleyan username.

Do not forget that your written work must be submitted as a PDF! Make sure that at the top of each file you have put your name! Do not, however, change the names of the files.