CS 5/7-344

Computer Networks & Distributed Systems

Fall 2023

Practice Final Exam

Student Name: _____

SMU ID: _____

Exam instructions

You are allowed the following resources for this exam:

- Two sheets of notes (A4 – both sides)

Please put away all laptops and cell phones. You must use your own resources for this exam – sharing of resources is **not** allowed.

WHEN YOU HAVE COMPLETED THE EXAM, PLEASE SUBMIT YOUR EXAM AND NOTE SHEETS. PLEASE PUT YOUR NAME ON BOTH ITEMS.

For instructor use only

Student Name: _____

SMU ID: _____

Problem	Points	Score
1	20	
2	20	
3	20	
4	20	
5	20	
6	20	
Total	100	

Any 5 of the 6 questions

TIME: 2 hours

Answer any 5 of the 6 questions.

1)

A. (5 points) Suppose your house has two rooms. In the first room, **2 computers** are connected with a **LAN**, and in the second room, **3 computers** are connected with another **LAN**. And both of the LANs are connected with a **router**. Can you draw this network?

B. (5 points) Write the NRZ (non-return to zero), NRZI (non-return to zero invert), and Manchester forms for the following bit stream.

10011100011

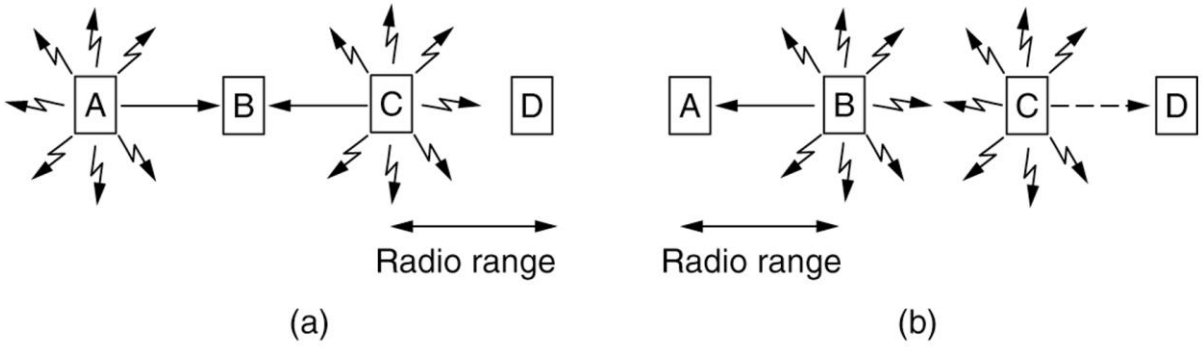
C. (5 points) The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC FLAG FLAG D. What is the output after stuffing?

D. (5 points) Explain Piggybacking.

2)

A. (10 points) How does pure ALOHA work? Why there is a chance of collision? Explain with a diagram.

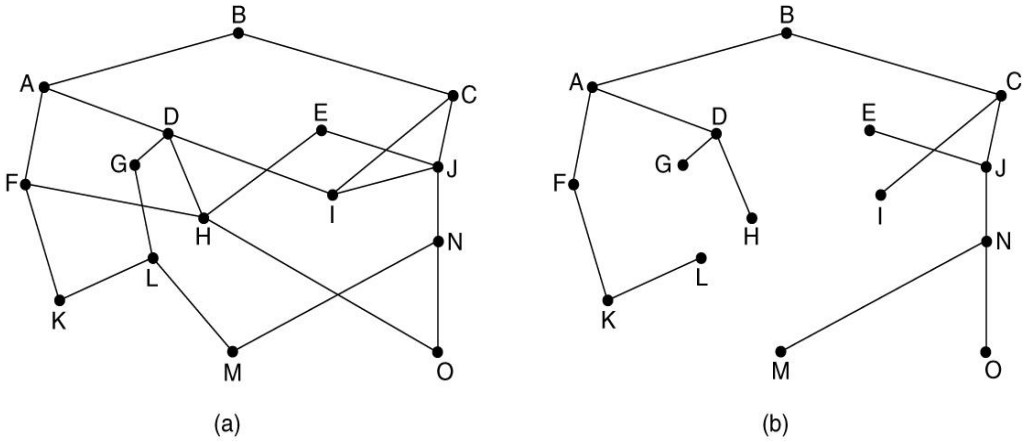
B. (10 points) What are hidden and exposed terminals? Explain with help of the following diagrams (a) and (b).



(a) (b)

3)

A. (10 points) What are optimality principle and sink tree? Define based on the following diagram.



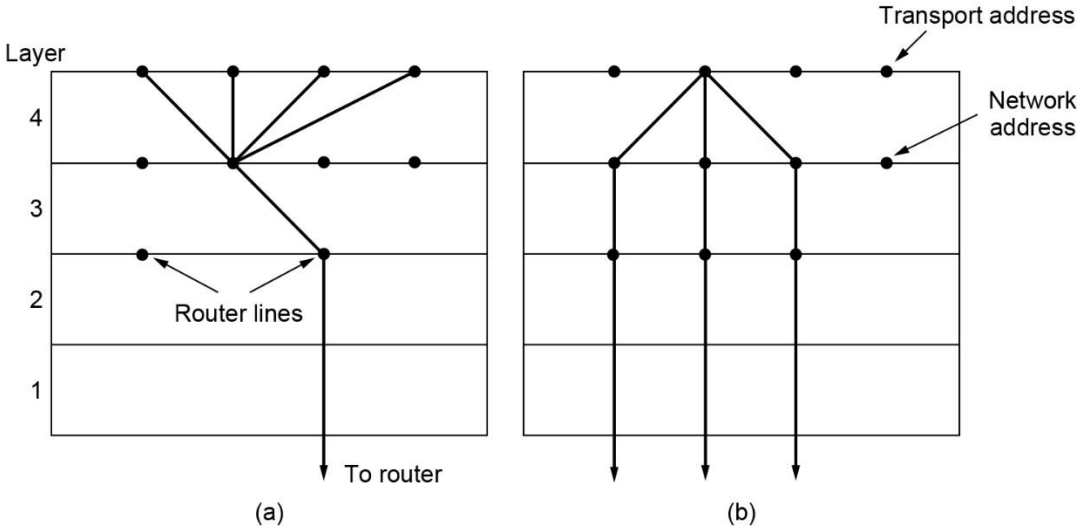
a) A network. (b) A sink tree for router *B*

B. (10 points) Write drawbacks of the following routing algorithms.

- Distance vector routing
- Link state routing
- Hierarchical routing within a network
- Broadcast routing

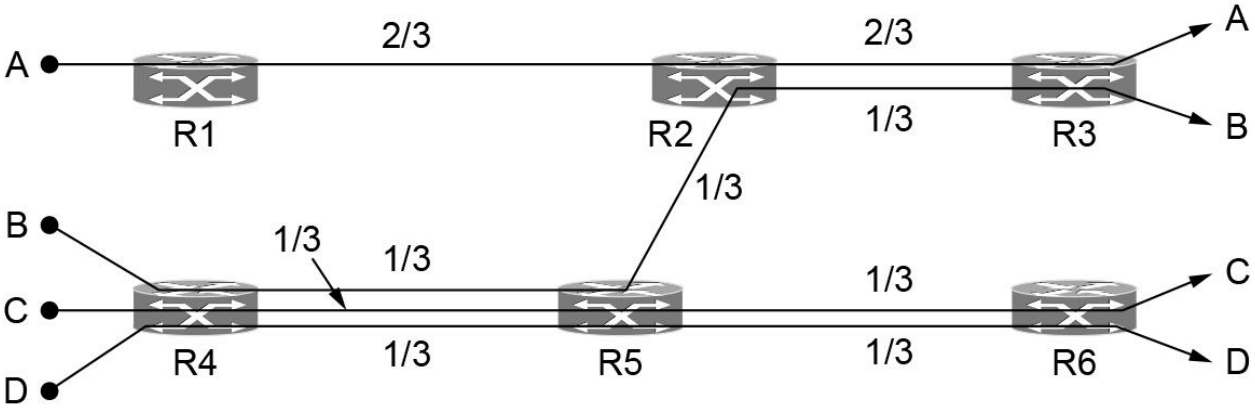
4)

A. (10 points) Explain multiplexing and inverse multiplexing according to the following diagram.



(a) Multiplexing. (b) Inverse multiplexing.

B. (10 points) Explain minmax fairness and use the following diagram as an example.



5)

A. (10 points) Find differences between the following-

- Difference between HTTP and HTTPS
- Difference between static and dynamic content or object
- Difference between HTTP/1 and HTTP/1.1
- Difference between HTTP/1.1 and HTTP/2

B. (10 points) How do we transfer digital video through network? Write each step.

6)

A. (10 points) For each of the following applications, tell whether it would be (1) possible and (2) better to use a PHP script or JavaScript, and why:

- Displaying a calendar for any requested month since September 1752.
- Displaying the schedule of flights from Amsterdam to New York.
- Graphing a polynomial from user-supplied coefficients.

B. Explain the following Client-side socket programming.

```
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#define SERVER_PORT 8080                    /* arbitrary, but client & server must agree */
#define BUF_SIZE 4096                       /* block transfer size */


int main(int argc, char **argv)
{
  int c, s, bytes;
  char buf[BUF_SIZE];                       /* buffer for incoming file */
  struct hostent *h;                        /* info about server */
  struct sockaddr_in channel;               /* holds IP address */

  if (argc != 3) {printf("Usage: client server-name file-name0); exit(-1);}
  h = gethostbyname(argv[1]);               /* look up host's IP address */
  if (!h) {printf("gethostbyname failed to locate %s0, argv[1]); exit(-1;}

  s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
  if (s <0) {printf("socket call failed0); exit(-1);}
  memset(&channel, 0, sizeof(channel));
  channel.sin_family= AF_INET;
  memcpy(&channel.sin_addr.s_addr, h->h_addr, h->h_length);
  channel.sin_port= htons(SERVER_PORT);
  c = connect(s, (struct sockaddr *) &channel, sizeof(channel));
  if (c < 0) {printf("connect failed0); exit(-1);}

  /* Connection is now established. Send file name including 0 byte at end. */
  write(s, argv[2], strlen(argv[2])+1);

  /* Go get the file and write it to standard output. */
  while (1) {
       bytes = read(s, buf, BUF_SIZE);      /* read from socket */
       if (bytes <= 0) exit(0);             /* check for end of file */
       write(1, buf, bytes);                /* write to standard output */
  }
}
```