14.01.20

09.30 - 11.30am

CMPU 3006 Client Serving Programming

Basement 3, Kevin Street

Programme Code: DT228, DT282
Module Code: CMPU 3006
CRN: 22404, 30085

TECHNOLOGICAL UNIVERSITY DUBLIN

KEVIN STREET CAMPUS

BSc. (Honours) Degree in Computer Science

BSc. (Honours) Degree in Computer Science (International)

Year 3

SEMESTER 1 EXAMINATIONS 2019/20

Client Server Programming

Mr. Damian Bourke
Dr. Deirdre Lillis
Dr. Martin Crane

Two Hours

Attempt all questions. NOT all questions carry the same mark.

1. In relation to HTTP Requests: **[20 marks]**

(i) The following string, representing an incorrectly formatted HTTP Request message, is sent to a HTTP/1.1 server. There are five errors associated with this string; four of which relate to syntax and one relating to an omitted compulsory data element. Rewrite the HTTP Request in its proper form.

GET /HTTP/1.1 \r\nConnection-closed\r\n

(5 marks)

(ii) Refer to Figure 1 - *A recv() Primitive*:

(a) Explain the purpose of the recv() primitive identifying the ***source*** buffer from which the incoming data stream originates. (5 marks)

(b) Assuming the test condition within the while loop is true i.e. numBytes > 0, explain what this means in terms of the status of the socket identified by the descriptor *clntSock*. (5 marks)

(c) Assume a properly formatted HTTP GET request is sent to a HTTP server containing this code. Explain the problem associated with this implementation of the *recv()* primitive and how this problem might be addressed. (5 marks)

```
    …..IGNORE ALL PRIOR LINES.

21      while ((numBytes = recv(clntSock, recvbuffer, BUFSIZE - 1, 0)) > 0)
22       {
23         recvbuffer[numBytes] = '\0';
24         fputs(recvbuffer, stdout);
25       }
26       if (numBytes < 0)
27         DieWithSystemMessage("recv() failed");
28
    …..IGNORE ALL SUBSEQUENT LINES.
```

Figure 1: *A recv() Primitive.*

2. Refer to Figure 2 (*A snippet of code*) and the following list of terms: **[20 marks]**

sock, bind, clntSock, connect, &servAddr, listen, servSock, clntAddr, socket, sizeof(sendbuffer), servAddr, accept.

(i) Using line numbers, identify which XXXX value can be replaced with which term from the above list. (12 marks)

(ii) State whether this is a client or server application justifying your answer. (4 marks)

(iii) If this snippet of code is derived from a *Daytime* application, identify the socket primitive that would be called after line 56. (4 marks)

```
    …..IGNORE ALL PRIOR LINES.

36  if (XXXX(servSock, (struct sockaddr*) &servAddr, sizeof(servAddr)) < 0)
37      DieWithSystemMessage("bind() failed");
38
39  if (XXXX(servSock, MAXPENDING) < 0)
40      DieWithSystemMessage("listen() failed");
41
42  for (;;) {
43
44      struct sockaddr_in clntAddr;
45
46      socklen_t clntAddrLen = sizeof(XXXX);
47
48      int clntSock = XXXX(servSock, (struct sockaddr *) &XXXX, &clntAddrLen);
49      if (clntSock < 0)
50        DieWithSystemMessage("accept() failed");
51
52      snprintf(sendbuffer, sizeof(sendbuffer), "%.24s\r\n", ctime(&ticks));
53
54      ssize_t numBytesSent = send(XXXX, sendbuffer, strlen(sendbuffer), 0);
55      if (numBytesSent < 0)
56        DieWithSystemMessage("send() failed");

    …..IGNORE ALL SUBSEQUENT LINES.
```

Figure 2: *A snippet of code.*

3. In relation to the concept of a ***socket*** discuss the following: **[15 marks]**

   (i) What is the purpose of a ***socket***? (5 marks)

   (ii) After creating a ***socket*** a server application modifies it to become a ***listening socket***. Explain how a *listening socket* differs from the socket that is created in a client application. (5 marks)

   (iii) At some point a ***connected socket*** is created within the server. Explain the purpose of a *connected socket* and explain how it differs from a *listening socket* in terms of: its lifespan i.e. how often it is created/destroyed during interactions with client applications and data exchange with client applications. (5 marks)

4. Refer to Figure 3 (*An internetwork*). **[15 marks]**

The addresses associated with each network entity are:

Host A: **Port 36408**, **Host A IP address** and **Host A MAC address**,
Host B: **Port 25**, **Host B IP address** and **Host B MAC address**,
Router: **net 1 MAC address**, **net 2 MAC address**, **net 1 IP address** and **net 2 IP address**.

(i) Identify the type of PDU required to move data across **net 1** and **net 2**. In your answer, and with reference to the addresses above, identify the source and destination addresses used in each case. (7 marks)

(ii) Identify the type of PDU required to move data from **Host A** to **Host B** (regardless of the underlying network architecture). In your answer, and with reference to the addresses above, identify the source and destination addresses used. (4 marks)

(iii) Identify the type of PDU required to move data from the Application (**appl.**) on Host A **to** the Application (**appl.**) on Host B. In your answer, and with reference to the addresses above, identify the source and destination addresses used. (4 marks)

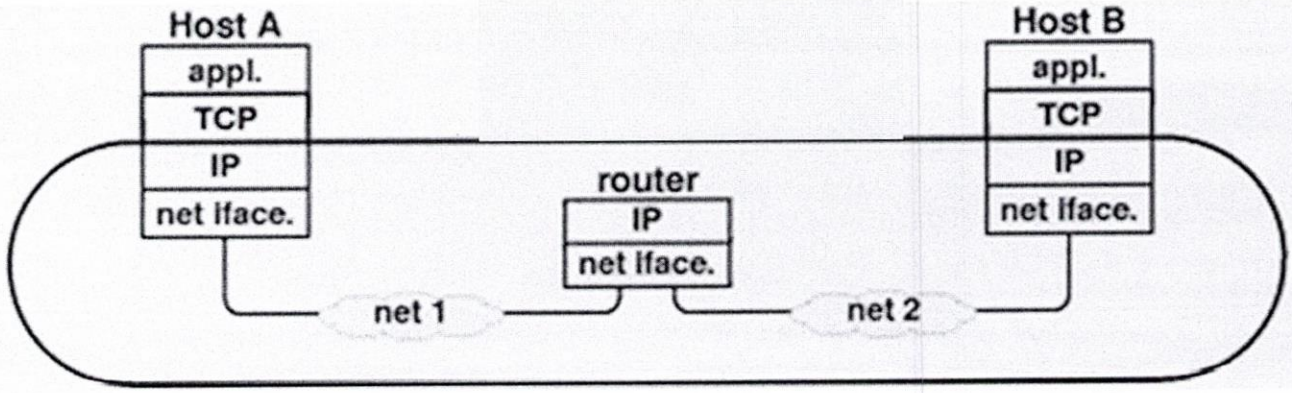

Figure 3: *An internetwork.*

5. Refer to Figure 4 (*An Email System*). **[15 marks]**

(i) Identify each of the Client-Server components labelled XXXX and YYYY and the Client component marked ZZZZ. (5 marks)

(ii) Explain the role of each of the following components: XXXX Client/Server, YYYY Client/Server and User Agent. (6 marks)

(iii) Identify the protocols used by the XXXX and YYYY Client/Server components and the ZZZZ Client component. (4 marks)

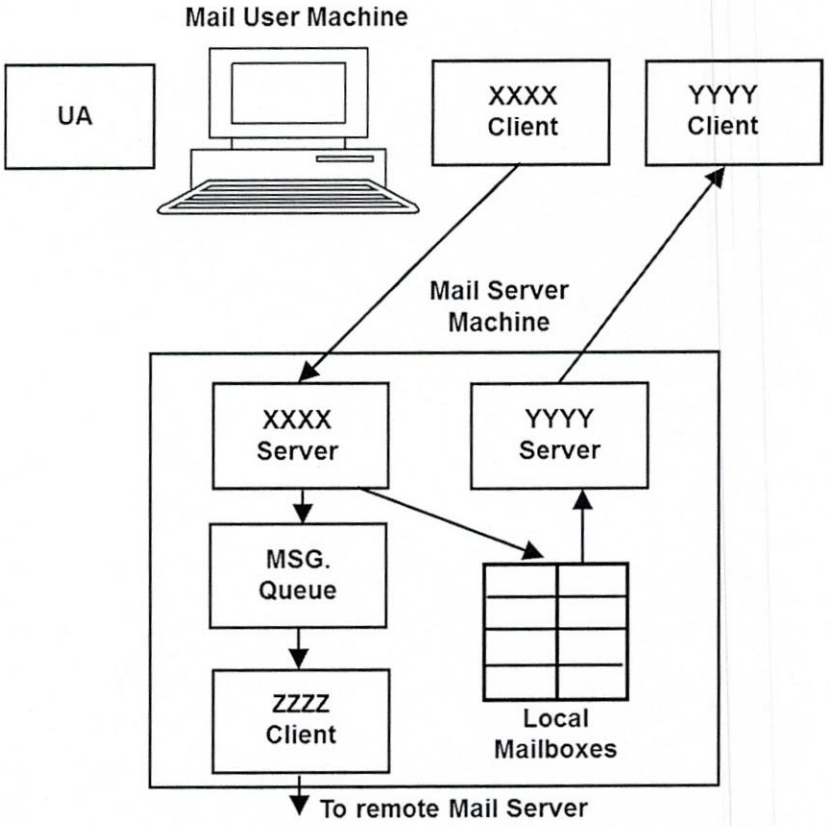

Figure 4: *An Email System.*

6. Refer to Figure 5 – *TCP State Transitions*. This figure shows TCP state transitions for the establishment of connections. **[15 marks]**

   (i) The terms *active* and *passive* are used to describe the process of connection establishment. Explain what these terms mean and identify which application, client or server, is normally associated with each term. (3 marks)

   (ii) Identify which transitions, dashed or solid lines, relate to each type of connection establishment (active or passive). (2 marks)

   (iii) Reproduce the diagram in your answer book. Identify on the diagram which of the following TCP message exchanges are used to move the status of the connection between each of the states.

   recv: SYN, send: SYN and ACK,
   send: nothing,
   send: SYN,
   recv: ACK, send: nothing,
   recv: SYN and ACK, send: ACK (10 marks)

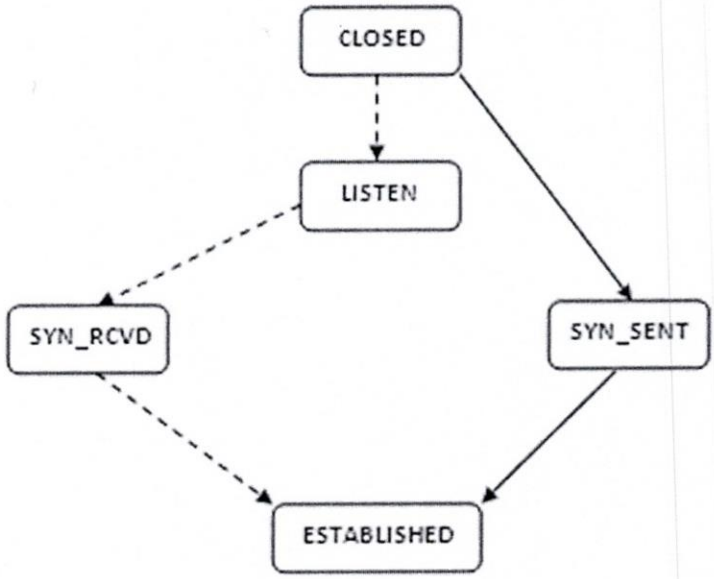

Figure 5 – *TCP State Transitions*.