

- 12. close(sockfd) always sends FIN on sockfd (T/F)
- 13. connect() can be used on raw sockets (T/F)
- 14. Raw sockets can be used for
 - (a) reading TCP packets
 - (b) reading UDP packets
 - (c) reading ICMP packets
 - (d) reading IGMP packets
- 15. If we need to watch on fds {5,3,9,40,10}, the first argument to select() will be
 - (a) 5
 - (b) 41
 - (c) 29
 - (d) 1024
- 16. epoll() can be used only for level-triggered notifications (T/F)
- 17. To put a timeout of 300 milliseconds on recvfrom() call expecting a reply, the following can be used
 - (a) select()
 - (b) poll()
 - (c) alarm()
 - (d) SO_RCVTIMEO socket option
- 18. The following can be used to send RST segment to a peer process in TCP
 - (a) SO_LINGER socket option
 - (b) send()
 - (c) shutdown()
 - (d) sendmsg()
- 19. In SUN RPC, there is a separate port number for every program (T/F)
- 20. To join a multicast group, a process must use the following socket option(s)
 - (a) IP_UNBLOCK_SOURCE
 - (b) IP_ADD_MEMBERSHIP
 - (c) IP_MULTICAST_IF
 - (d) IP_MULTICAST_TTL

Q2) Write code segments for the following. Syntax issues are not penalized. 3.5M

- 1. Put a timeout of 200 mille seconds on a udp socket. 2M

|

- 2. There are 5 FDs in a process which keep sending data at irregular intervals. Code should detect which fd has data and send it to all other fds.2M

|

Q3) Write protocol file for a function addAll() which accepts an array of integers and adds and returns integer. 2M

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
II SEMESTER 2019-2020
Comprehensive Examination

Course No.: IS F462	Course Title: Network Programming
Date: 10th May 2022 (8-11)	Maximum Marks: 35
Exam Type: Part B (Open Book)	Part B Marks: 25

Note:

- Write all parts of a question together.
-

Q1) Consider that a TCP client receives multiple IP addresses from a DNS server for a given domain name. TCP client finds out the IP address which is closest to it in terms of response time, and sends request. TCP client finds out closest IP concurrently for all given IPs so as to minimize time. Implement a complete TCP client for the above requirements.

[5]

Q2) Proxy server resides on an organization network. Web clients connect to proxy server first. Proxy server finds that if it has the content in cache, it will send to client otherwise it will connect to remote server to fetch content. Proxy server accepts client requests concurrently using I/O multiplexing approach. Implement a complete TCP based proxy server for the above requirements

[5]

Q3) Develop a multicast application for the following requirements.

- A member joins multicast group which uses 239.0.0.1 and 8888 as the port and sends its pid as the message.
- If pid of the sending and receiving process are odd numbers, receiver sends a TCP message to the sender on port 9000.
- If pid of the sending and receiving process are even numbers, receiver sends a UDP message to the sender on port 9000.
- Processes use unicast channels for private chatting. Chatting should support sending and receiving messages from multiple clients at the same time.

[5]

Q4) Implement a piece of code that retrieves the destination address of a packet which a UDP client sends to multi-homed UDP server.

[3]

Q5) State if the following are True or False. Give Justification.

- (a) In LAN, any host can receive multicast packets provided it binds the socket to the Multicast group address
- (b) Two traceroute programs running at the same time in a system can differentiate incoming packets based on source and destination IP addresses.

[2]

Q6) Consider the following paragraph given in section 16.5 of the textbook. We provide this example using simultaneous connects because it is a nice example using nonblocking I/O and one whose performance impact can be measured. It is also a feature used by a popular Web application, the Netscape browser. There are pitfalls in this technique if there is any congestion in the network. Chapter 21 of TCPv1 describes TCP's slow-start and congestion avoidance algorithms in detail. When multiple connections are established from a client to a server, there is no communication between the connections

at the TCP layer. That is, if one connection encounters a packet loss, the other connections to the same server are not notified, and it is highly probable that the other connections will soon encounter packet loss unless they slow down. These additional connections are sending more packets into an already congested network. This technique also increases the load at any given time on the server.

Consider the underlined lines. It tells about the missing coordination among the TCP connections or clients accessing the same web server. Design a solution that can enable this coordination considering the fact that TCP doesn't tell the application immediately about the packet loss. Generally, TCP tries for a few times before it concludes about packet loss. Your design should be able to detect packet losses as soon as the host comes to know about it and notify the client who are accessing that server. Your program should take care of accepting requests from clients, detecting packet losses, and notifying the clients. Implement the solution designed (only for the server which detects packet losses).

[5]

--&--