Questions

12. (a) Why do IP packets bounce back several times in the network? **[1]**

12. (b) Why a single RPC is required to be split over multiple packets? Does it by self or forceful to the system? Justify. **[1]**

12 (c) During a remote procedure call between client and server, why does copy of a message copy is required in different places? What is the minimum and maximum number of copies needed for a single communication? Mention the list of copies. How can we eliminate unnecessary copies during RPC? **[2.5]**

12 (d) Define a critical path in a remote procedure call. Mention all the steps in sequential order in the RPC from client to server. Analyze the amount of time wastage at different points of the critical path. **[2.5]**

Sample Answer

Answer: 12. (a): IP is not an end-to-end protocol. It is executed on top of TCP. As TCP is a connection-oriented reliable protocol. Due to different kinds of failures, the IP packets are required to be retransmitted across the network.

Answer: 12. (b): In a distributed environment, the packet and message length are varying. The varying packet size and message length are not supported by both the protocol and the network. Hence RPC mechanism is forced to be split over multiple packets, is an overhead.

Answer: 12 (c): Copy is an issue in RPC because the time consumed for copy exceeds it's actual execution time. The address space of the user and the kernel are different, so to keep track of the communication copy of the message is required at different places along the communication path. The minimum number of copies is 5, and the maximum number of copies would be 8. The list of copies are:

i. The network chip can DMA the message directly out of the client stub's address space onto the network.
ii. If(the kernel can't map the page into the server's address space) then the kernel copies the packet to the server stub.

iii. The hardware is started, causing the packet to be moved over the network to the interface board on the destination machine
iv. When the packet arrives, an interrupt occurs, and the kernel copies it to its buffer (before knowing its exact location)
v. Finally, the message has to be copied to the server stub
vi. If(the call has a large array passed as a value parameter) the array has to be copied onto the client's stack for the call stub,
vii. Copy from the stack to the message buffer during marshaling within the client stub
viii. Copy from the incoming message in the server stub to the server's stack preceding the call to the server.

An unnecessary copy can be eliminated by using the hardware "scatter-gather". By using suitable hardware, a reusable packet header inside the kernel and a data buffer in user space can be put out onto the network can avoid copying at the sender side. Similarly, if the whole message is being dumped into the kernel buffer of the receiver side, then copying at the receiver side would not be required. The use of virtual memory can avoid copying carried out by the operating system.

Answer: 12 (d): The sequence of instructions that is executed on every RPC is known as the critical path. There are 14 steps in the RPC from client to server. The steps are as follows:

i. Call stub
ii. Get message buffer
iii. Marshal the parameters
iv. Fill in headers
v. Computation of the UDP checksum
vi. Trap to the kernel
vii. Queue the packet for transmission
viii. Move the packet to the controller over the QBus
ix. Ethernet transmission time
x. Obtaining the packet from the controller
xi. Interrupt service routine
xii. Computer UDP checksum
xiii. Context switch to the user space
xiv. Server stub code

During the critical path, most of the time spent is – marshaling the parameters, moving the packets to the network interface for transmission, and managing a pool of buffers.