Subject: Solution of 10 networking questions

Client: (Deleted for confidentiality)
Submitted by: Siddharta Saha

Deadline: 30 hours
Actual time taken:14 Hrs 30 Mins.

- **Q1.** Not every node in a peer-to-peer network should become super-peer. What are reasonable requirements that a super-peer should meet?

Ans: Selection of super peer in peer-to-peer network should be done depending on following criterion

a. Availability. Since all the other peers within a cluster communicate via it so a super peer should have a very high availability.
b. Latency to access such node from any member of the cluster should be low.

- **Q2.** In a structured overlay network, messages are routed according to the topology of the overlay. What is an important disadvantage of this approach?

Ans: By definition "An overlay network is a virtual network built on top of a physical (underlay) network." While nodes in underlay network are essentially routers but in overlay network node may be a router, a host, a server or even an application.
When a message is routed across such network the shortest path between source and destination is chosen. But this logical shortest path many not be the physical best path. The source and receivers may be logically next to each other but physically at the remotest part of the network. So it can unnecessarily delay the message delivery even the other better path is available. Peer to peer applications are example of overlay networks.

- **Q3.** In the text, we described a multithreaded file server, showing why it is better than a single-threaded server and a finite-state machine server. Are there any circumstances in which a single-threaded server might be better? Give an example.

Ans: In most of the cases multithreaded servers are better than single threaded. But at some of the cases opposite holds good. For example say in case of spreadsheets changing a single field may result to change several other fields and thus a large number of computations is needed before this call is released.

Q4. Many networked systems are organized in terms of a back office and a front office. How do organizations match with the coherent view we demand for a distributed system?

Ans: In the real world distributed system is implemented as per the functional units of the organization. Such system does not have to necessarily have to span across the entire territory of the organization uniformly. So an organization have one distributes system for back office, one for front office and many others if other functional units are required within the organization. These systems can be amalgamated together or kept independent depending on the requirement and the policy of the organization. Example Windows 2003 server and application platform.

- **Q5.** Constructing a concurrent server by spawning a process has some advantages and disadvantages compared to multithreaded servers. Mention a few.

  Ans:

- **Q6.** What is the difference between a vertical distribution and a horizontal distribution?

Ans:

Vertical distribution	**Horizontal distribution**
Different layers of the distributed process are distributed across different nodes. Each layer is implemented in one node. Example DB is kept in one server and corresponding web server is kept else.	

- **Q7.** An alternative definition for a distributed system is that of a collection of independent computers providing the view of being a single system, that is, it is completely hidden from users that there even multiple computers. Give an example where this view would come in very handy.

Ans : The ideal example for it would be bank ATM system. A customer comes to the ATM, inserts his ATM card provides credentials and withdraws money. To him it is just a machine that delivers cash. But the actual system is complex and distributed amongst several servers/ machines. There machines are independent and redundant but that fact is never understood by the user i.e. customer.

- **Q8.** Explain what is meant by a virtual organization and give a hint on how such organizations could be implemented.

Ans:

- **Q9.** If a client and a server are placed far apart, we may see network latency dominating overall performance. How can we tackle this problem?

Ans: Latency is problem of any communication. In client server architecture performance of clients depend greatly on the latency. When the latency is high clients would experience delay this may be improved I following methods

- Instead on large request-replies client side codes could be broken into smaller parts. So that a when small amount of data is received the client can start work instead of waiting for a bigger chunk of data. In the meanwhile next piece of code can be scheduled for work.
- Clients can run multiple sessions with the server and data obtained can be added up at client end (e.g. Internet download Manager).
- The client can utilize the delay time between sending of request and receiving reply for other processes.

- **Q10.** Would it make sense to limit the number of threads in a server process? Why?

Ans: For all practical implementation it is a best practice to limit the number of threads in a server process.