1. **Explain the limitations of Transmission Control Protocol in a high throughput networking environment.**

   TCP/IP has a fixed number of bits for representing the window size. For example, In IPv4 it has 16 bits for the window size, which means 65535 possible sizes, implying that a maximum of 65535 bytes can remain unacknowledged at any given instance between the client and the server. Even though the link may have capacity to carry more data, it will not transmit until it gets the acknowledgement back from the receiver. This causes TCP/IP to be wasteful of the high speed/bandwidth available to it.

   Also, in TCP/IP , before transmission starts occurring, a three-way handshake needs to be done in order to set-up a connection between the client and the server. This requires some time, though not much but can impact the speed, say if the message was supposed to be just one packet, but would have to wait for the 3-Way Handshake to complete.

2. **Compare the Stream Control Transmission Protocol and FastTCP.**

   SCTP:
   - Is a reliable transport-layer protocol as an overlay on a connectionless protocol such as IP; Attempts to bridge the gap between TCP and UDP
   - Message oriented, uses associations rather than connections
   - Reliable Transfer, unreliable connection
   - Supports multi-streaming and multi-homing
   - 4-Way Handshake, making it SYN-attack resistant
   - Slow start and congestion avoidance phases like TCP
     - Selective Acknowledgement
     - Fast Retransmit
   - Partial data ordering
   - Uses rwnd, cwnd and ssthresh

   FastTCP:

   - Overcomes congestion control shortcomings of TCP
   - Quicker response to network congestion
   - Queuing delays and packet loss to control congestion
   - 3-Way Handshake

3. **Explain the differences between the three basic forms of Inter-Process Communications (IPC). Which cannot be operated over a network, and why?**

   Three types:
   - **Shared Memory**
     In shared memory model, all workers/processes operate on the same data. Since there is one single memory being used by all, this can't *cannot be operated over a network*.
   - **Message Passing**
     Message passing systems allow workers/processes to communicate through a messaging system. Messages keep everyone separated, so this *can occur over a network*. So processes on different hosts can actually work together without sharing physical memory.

- **Unix Signals**
  Unlike the previous two, this relies on the system kernel for processes sending signals to other processes. The signals co-ordinate communication between the processes and hence form IPC. Since this requires one common kernel, this usually *does not happen over a network*.

4. **Explain the internal and functional differences between the Berkeley socket scheme, and the AT&T System V STREAMs scheme.**
   - Both Berkeley Socket scheme and AT&T System V STREAMs have got their own APIs to for socket control.
   - Berkeley Sockets are implemented using a library of linked modules. Whereas in System V (SysV) it is implemented as a chain of coroutines that pass messages between a program and a device driver. SysV has many other implementations available as well.
   - Berkeley Sockets is an API specifically written in C, with a nice wrapper for ease of use. STREAMs on the other hand is not written in any programming language, and instead works with API for the Unix kernel on the command line.