Question #1:

The minimum seek time for an HDD is 9 msec, and the maximum seek time is 90 msec. The block size of this HDD is 4KB. How long on average does it take to read 100MB of data?

Answer:

Key components are:

* Seek Time
* Rotational latency
* Command processing time
* Settle time.
* Transfer Time
* Data transfer rate

Access Time – measure of the time it takes before the drive can transfer data… factors are related to mechanical nature of rotating disks.

Seek: A rotating drives average seek time is the average of all possible seek times which technically is the time to do all possible seeks divided by the number of all possible seeks. In reality, The average seek distance on a disk over all possible seeks is calculated as 1/3 rd of the full distance.

Rotational Latency – is the delay waiting for the rotation of the disk to bring the required disk sector under the read-write head.

Command processing time– time to set up communication… time is typically not included.

Settle Time – also not included

Data Transfer Rate – aka throughput… covers both internal rate and external rate (internal is moving data between disk surface and the controller on the drive and external is moving data between the controller on the drive and the host system.

Read time is calculated as: (Seek Time + Rotational Latency + Transfer Time) (assuming 200rpm)

9ms + ((1 / (200 / 60) \* 0.5 \* 1000) + ~ instant …. == ~ 150 ms

Question #2 –

Describe a TCP/IP packet in detail. Describe the header, how many bytes it is, and which components it contains. What data can come after the header?

Answer:

**TCP**/**IP** stands for “Transmission Control Protocol / Internet Protocol”. It is basically a network protocol that defines the details of how data is sent and received through network adapters, hubs, switches, routers and other network communications hardware.

TCP use headers as part of packaging message data for transfer over network connections. Each TCP header has ten required fields totaling 20 byes (160 bits) in size

TCP headers:

1. Source TCP port number (2 bytes) *Source and destination TCP port numbers* are the communication endpoints for sending and receiving devices.
2. Destination TCP port number (2 bytes)
3. Sequence number (4 bytes) Message senders use *sequence numbers* to mark the ordering of a group of messages. Both senders and receivers use the acknowledgment*numbers* field to communicate the sequence numbers of messages that are either recently received or expected to be sent.
4. Acknowledgment number (4 bytes)
5. TCP data offset (4 bits) - The *data offset field* stores the total size of a TCP header in multiples of four bytes. A header not using the optional TCP field has a data offset of 5 (representing 20 bytes), while a header using the maximum-sized optional field has a data offset of 15 (representing 60 bytes).
6. Reserved data (3 bits) *Reserved data* in TCP headers always has a value of zero. This field serves the purpose of aligning the total header size as a multiple of four bytes (important for efficiency of computer data processing).
7. Control flags (up to 9 bits) TCP uses a set of six standard and three extended *control flags*(each an individual bit representing *on* or *off*) to manage data flow in specific situations. One bit flag, for example, initiates TCP connection reset logic. The detailed operation of these fields goes beyond the scope of this article.
8. Window size (2 bytes) TCP senders use a number called *window size*to regulate how much data they send to a receiver before requiring an acknowledgment in return. If the window size becomes too small, network data transfer will be unnecessarily slow, while if the window size becomes too large, the network link can become saturated (unusable for any other applications) or the receiver may not be able to process incoming data quickly enough (also resulting in slow performance). Windowing algorithms built into the protocol dynamically calculate size values and use this field of TCP headers to coordinate changes between senders and receivers.
9. TCP [checksum](https://www.lifewire.com/what-does-checksum-mean-2625825) (2 bytes) The *checksum*value inside a TCP header is generated by the protocol sender as a mathematical technique to help the receiver detect messages that are corrupted or tampered with.
10. Urgent pointer (2 bytes) The urgent pointer field is often set to zero and ignored, but in conjunction with one of the control flags, it can be used as a data offset to mark a subset of a message as requiring priority processing.
11. TCP optional data (0-40 bytes) Usages of optional TCP data go beyond the scope of this article but include support for special acknowledgment and window scaling algorithms.

Question #3:

How does the network protocol guarantee that a TCP/IP packet is complete after transmission?

Answer:

Each packet includes the source and destination address, which helps in delivering packets successfully but if some packets are missing then information can be sent and the packet gets retransmitted. The header information describes the version, length of packet, checksum, and other information which helps in reconstruction and correction of message. It also make the data transmission fast and reliable.

It originated in the initial network implementation in which it complemented the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP). Therefore, the entire suite is commonly referred to as [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP). TCP provides [reliable](https://en.wikipedia.org/wiki/Reliability_(computer_networking)), ordered, and [error-checked](https://en.wikipedia.org/wiki/Error_detection_and_correction) delivery of a stream of [octets](https://en.wikipedia.org/wiki/Octet_(computing)) between applications running on hosts communicating by an IP network. Major Internet applications such as the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), [email](https://en.wikipedia.org/wiki/Email), [remote administration](https://en.wikipedia.org/wiki/Remote_administration), and [file transfer](https://en.wikipedia.org/wiki/File_transfer) rely on TCP

Question #4:

What is the difference between TCP and IP?

Answer:

The difference is that TCP is responsible for the data delivery of a packet. TCP is a connection oriented and reliable protocol… it provides the delivery of a stream of data from one computer to another. Controls most of the flow, size, and rate.

IP is responsible for the logical addressing. IP obtains the address and TCP guarantees delivery of data to that address. IP is a network layer protocol, it is a connection less protocol. Only provides addresses for transmission.

Question #5:

Why is 3d performance so much higher with a graphics card than without? Modern CPUs are extremely fast, what is limiting their performance?

Answer:

GPU-accelerated computing offloads compute-intensive portions of the application to the GPU, while the remainder of the code still runs on the CPU. From a user's perspective, applications simply run much faster.

A CPU consists of a few cores optimized for sequential serial processing

A GPU has a massively parallel architecture with thousands of cores designed for handling multiple tasks simultaneously.

Specific to graphics and 3d performance having an additional card that optimizes performance heavy matrix calculations.

The CPU continues to power everything else but is helped by granting the CPU less stress and avoiding heat issues.