1. Congestion control is implemented in TCP with a sliding window technique. A maximum send speed is advertised by the receiver, and the sender sends data at this rate. Once the data is acknowledged, the sender moves along the bit stream, sending data at this speed. By using timers the communicating hosts can keep updated round-trip time figures and determine if a packet was lost. If timeouts occur, the data is resent with a longer timer wait. The standard algorithms for congestion control in TCP are Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery. In slow start, the hosts agree on a maximum segment size. With each successful send and ack, the maximum segment size is increased by one, until a timeout occurs, determing optimal maximum segment size. Once a timeout is observed in slow start, the Congestion Avoidance algorithm slow the transmission rate. Once congestion is observed by an expiring retransmission timer or the reception of duplicate ACKs, the Congestion Avoidance algorithm reduces the maximum segment size by one half, or to 1. If the congestion was flagged by a timeout, segment size is set to one the client put back to slow start. If congestion was indicated by duplicate ACKs, the Fast Retransmit and Fast Recovery algorithms are invoked. Fast Retransmit is used in the case of three or more duplicate acks at the sender, and the sender bypasses any retransmit timer and retransmits, thus Fast Retransmit. In Fast Recovery, rather than revert to Slow Start mode, the sender retransmits with a larger window, increasing window size as if in Congestion Avoidance mode.
2. When a user clicks a web link, the link location is resolved to a host, querying DNS for an IP and directly connecting to the IP at the given port number. Once the connection is established, the requested file is downloaded to the user’s local cache. To improve this process, several techniques are used, like CDNs. Content Delivery Networks use redundant servers spread out along a network to obtain maximum bandwidth.
3. Fingerprinting is the process by which information can be obtained about a remote host by simply observing how it responds to different types of traffic. There are several parameters within the TCP protocol that are not universally defined, leading to varying values for these parameters based on implementation. Simply recording the types of messages moving to and from the machine can reveal the TCP/IP implementation and host OS. Active fingerprinting involves the offending machine to initiate contact, perhaps with a ping or IMCP timestamp request, using the format to determining information about the host. Passive fingerprinting is fingerprinting by watching traffic going out from a machine to other machines to glean information. A good defense against fingerprinting to limit the amount of traffic to/from the machine, by blocking unneeded types.
4. Ensuring QoS on the internet can be done is several ways. For example, traffic shaping. Traffic/packet shaping increases performance and bandwidth by delaying packets that meet certain criteria. Packets are categorized into categories based on type/importance and prioritized appropriately by throttling either bandwidth or transfer rate. Two examples of methods used in this type of QoS are overflow condition and traffic classification. The overflow condition is used to invoke traffic shaping by simply dropping any new packets once a buffer is full, causing the network to automatically fall back to it throttling mechanism. Traffic clarification is prioritizing traffic based on type. I.E. CAD data gets sent before email. Another method of bandwidth management is the scheduling algorithm. In the scheduling algorithm, traffic for each destination is divided and sent over multiple paths. An example the fair queuing algorithm, where data is forwarded from a buffer, which acts as a queue, where the packets are stored until transmission. Fair queuing works to achieve max-min fairness, where it first works to maximize the minimum data rate of any active data flow, and then the second minimum data rate, the third, etc. This type of scheduling algorithm results in lower throughput than maximum throughput scheduling, where the lowest-cost data flows are granted highest priority in an attempt to maximize the total throughput of the network. Congestion avoidance algorithms works much like the sliding window technique for flow control. Once congested has been detected, congestion control works to reduce resource usage. One way this done is to reduce the rate of sending packets. Similar to TCP’s sliding window, once the appropriate number of duplicate acks are received, the congestion avoidance algorithm can use techniques such as Tahoe and Reno. In Tahoe, loss is detected by an expired timeout before an ACK is received. The congestion window is then reduced to 1 MSS, and the connection is set to slow-start state. In Reno, the congestion is reduce by one half on the receipt of three duplicate acks. This triggers a fast retransmit, then Fast Recovery mode is enabled, in which the missing packet is retransmitted and the connection returns to congestion avoidance mode. If this retransmission times out, the connection goes to slow-start mode.
5. The data link layer does things like: error control, media access control, data encapsulation, and VLANs. The transport layer is responsible for flow control, congestion avoidance, ports, and also some error control (ensures the reliable transfer of data). A transport layer connection is a temporary logical connection that closes with the process finishes. The job of the DLL is to provide upper layers a medium that acts as though it is error free. This is done by dividing input into frames and sending the frames in order. Essentially, the DLL is responsible for sending logical packets over the physical layer, and to provide some form of error control (in the form of acks, etc.). The transport layer is responsible for delivering packets to processes on remote hosts, including the handling of ports for the session layer. The DLL is basically the network card driver, which serves as a software interface for the transport layer. Essentially, the main difference is that the DLL creates frames and tries to provide a perfect “data link”, while the transport layer (a higher level) creates packets to route data and provide QoS. The technical reason for this is similar to the reason why the model is layered. The goal of the DLL is to abstract the hardware so the transport layer works despite hardware changes.
6. To build a network to support this organization, I would place related departments physically next to each other and run one network to that area (scalable as the building size/utilization increases), install a switch of sufficient size (scalable) to support the users in that area, use wireless (scalar) to bridge coverage gaps (saturate building with wireless, scalable, allows laptops where PCs cannot go), use a unicast (or a complex multicast) network , create VLANs for related departments to implement QoS priority along the same/certain VLAN. Use a VPN on T1 line(s) (scalable) for expansion to other buildings. For security, a Windows Server with Active Directory, a software firewall and strong group policy should be used. Servers with specific roles can be added later (scalable).
7. MP3s use p**erceptual noise shaping to discard extraneous information with regard to the human ear; noises that are outside of the audible range for humans is stripped. Likewise, when the human ear hears two sounds, only the louder is perceived by the brain, so the quieter one is discarded in MP3. The amount of information that is discarded (or the aggressiveness of the compression algorithm) is determined by the user-set bit rate. For example, to record a one minute song in CD quality, you need a minimum sample rate of 44.1 kHz in a 16 bit format.** That is 44100 values per second of sound. Multiply this by 2 for stereo, and multiply this by two again since we use 2 bytes (16 bit). This is how data is first encoded into a digital “bit stream” before being compressed. So the size of 1 minute of music is 44100 samples/second \* 2 channels \* 2 Bps \* 60 seconds ~ 10 MB. To compress the data, the encoded sounds are translated into mathematical expressions and processed as outline above, using techniques from the study of psychoacoustics.
8. (a)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| To | B | D | G | A | Thru |
| A | 6 | 4 | 4 | 0 | A |
| B | 0 | 5 | 5 | 3 | B |
| C | 3 | 2 | 9 | 6 | B |
| D | 5 | 0 | 6 | 7 | D |
| E | 6 | 6 | 10 | 9 | B |
| F | 9 | 3 | 3 | 8 | G |
| G | 11 | 4 | 0 | 5 | G |
| Delay | 3 | 7 | 5 | -- | -- |

(b)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | **M** | DA | DB | DC | DD | DE | DF | DG |
| 1 | **A** | **0A** | ∞ | 4A | ∞ | ∞ | ∞ | 2A |
| 2 | **A, G** | -- | ∞ | 4A | ∞ | ∞ | ∞ | **2A** |
| 3 | **A, G, B** | -- | **3G** | ∞ | ∞ | ∞ | 8G | -- |
| 4 | **A, G, B, C** | -- | -- | **2B** | ∞ | 5B | ∞ | -- |
| 5 | **A, G, B, C, E** | -- | -- | -- | 2C | **1C** | ∞ | -- |
| 6 | **A, G, B, C, E, F** | -- | -- | -- | ∞ | -- | **4E** | -- |
| 7 | **A, G, B, C, E, F, D** | -- | -- | -- | **3F** | -- | -- | -- |

(c)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| To | B | C | G | A | E | New B | Line |
| A | 4 | ∞ | 2 | 0 | ∞ | 2 | G |
| B | 0 | 2 | 3 | 4 | 5 | 0 | - |
| C | 2 | 0 | ∞ | ∞ | 1 | 1 | E |
| D | ∞ | 2 | ∞ | 4 | ∞ | 2 | C |
| E | 5 | 1 | ∞ | ∞ | 0 | 1 | C |
| F | ∞ | ∞ | 8 | ∞ | 4 | 4 | E |
| G | 3 | ∞ | 0 | 2 | ∞ | 3 | G |

1. (a) Assume 32-bit sequence numbers;   
    232 sequence numbers = 4294967296 B  
    400Mbps = 50 MBps = 52428800 Bps / 4294967296 B = 0.01220703125 seconds before the sequence number wraps around
3. Al's needs 250, so give two Class C subnets for a total of 512 address, mask: 192.24.16.0/23 IT needs 1000, so give eight Class C subnets for a total of 2048 address, mask: 192.24.16.0/21   
   U-BB needs 5000, so give 20 Class C subnets for a total of 5376 address, mask: 192.24.16.0/19   
   Joe's needs 100, so give one Class C subnets for a total of 256 address, mask: 192.24.16.0/24 using the starting address of 194.24.16.0, and the subnet mask 255.255.224.0 you have a maximum of 8190 address. Thus, assign as follows (accounting for considerable growth):  
   Al's : 194.24.16.0 through 194.24.16.255  
   IT : 194.24.17.0 through 194.24.24.0  
   U-BB : 194.24.25.0 through 194.24.26.0  
   Joe's : 194.24.27.0 through 194.24.46.0