367 Assignment 1

Matt Forbes

January 12, 2011

1.4

Digitized Voice Traffic

Digital voice networks should provide a QoS that is reliable in the sense that it won't drop connections, and must also relay data between machines in realtime or close to it. If there was too much of a delay between senders/receivers, then nobody could hold a conversation.

Video Traffic

Videos are normally huge, so a network streaming video should be able to handle large transfers at high speed. Video is normally played at 20-30 frames per second, so small errors in the data would not be noticed, and high speeds would make up for errors in picture. So QoS is defined as high speed with acceptable errors in playback.

Financial Transaction Traffic

Accuracy and security of financial transactions are key. Packets sent to/from a bank's server should be unreadable by everyone but the intended recipient. Further, the data sent needs to be exact, an extra zero can cause a whole lot of damage.

1.10

Layered networks provide abstraction that is necessary to cope with changes and expansion of hardware. Moving up from the physical layer, each step up moves further and further from physical details and plumbing. A developer writing a HTTP server doesn't want to deal with the details of writing code that transports bits to client, handling special cases for each type of hardware. Using a layered network, this coder can simply offload this work to a lower level and let it take care of the details. Layered networks can add a lot of overhead when the data transmitted is really small. As a packet moves between layers, it accumulates headers along the way which can add up to more than the original data.

1.13

When there are certain parameters that can change the way the network can handle a connection, all parties involved can negotiate what these options will be. An example might be a client connecting to a server that is going to transfer it file. When they are connecting, they would define the details of the transaction like maximum packet size, maximum file size, etc.

1.20

These two approaches seem to be concerned with the reliability of the connection. On one hand, acknowledging every single packet might seem a little overboard, but when a packet doesn't make it through, it will be easy for the sender to simply resend that specific packet. But these extra acknowledgements might add up to a lot of overhead if it isn't crucial to send all the packets.

When only acknowledging when the whole file is sent, it may be a lot cheaper depending on the size of the file and total number of packets. If the sender never hears back from the receiver, there will be no way of simply sending a missing packet, the whole operation has failed and would have to be restarted or aborted.

1.23

1600 x 1200 pixels is 1920000 pixels total. If each pixel is 3 bytes, then there are 5760000 bytes or 46080000 bits total.

1kb = 1024 bits, so 56kbps = 57344bps, so downloading this file would take $\frac{46080000}{57344} = 803$ seconds.

1Mb = 1048576 bits, so 1Mbps = 1048576bps, so it would take $\frac{46080000}{1048576} = 43$ seconds to download this file.

100Mb = 104857600 bits, so 100Mbps = 104857600bps. It would take $\frac{46080000}{104857600} = 0.44$ seconds.

1Gb = 1073741824 bits, so 1Gbps = 1073741824bps. It would take $\frac{46080000}{1073741824} = 0.0429$ seconds.