1.4) For the purposes of this problem, we will define the following variables:

F = file size
P = packet size
L = link bandwidth

2 points for a completely correct answer, 1 point for minor errors, 0 points for a blank answer.

Many of you had problems with unit conversion. If you had unit conversion problems and still showed your work and had the right idea otherwise, we only counted you off for it once.

a. Since data packets can be sent continuously, it takes F / L time to send the entire file and an additional 0.5 RTT for the last byte of the file to propagate from the sender to the receiver. Adding in the additional 2 RTT handshake time, we arrive at

2.5 RTT + F / L

or 1.458 seconds to perform the entire transfer.

b. We need to send F / P = 1536 packets. From the sender's perspective, we must wait 1 RTT between sending each packet (F/P - 1 times), and wait an additional 0.5 RTT for the last byte of the last packet to propagate from the sender to the receiver. It takes P / L time for the sender to transmit each packet. Each packet takes 0.5 RTT to propagate through the link from sender to receiver, but we don't have to consider that time because it occurs concurrently with and is shorter than the sender's 1 RTT delay. Factoring in the initial 2 RTT handshake, we arrive at

2.5 RTT + [(F / P) - 1] * [(P / L) + 1 RTT]
= 2.5 RTT + (F / L) + [((F / P) - 1) * 1 RTT]

= 124.258 seconds

Note that this is the solution to a. with an additional ((F / P) - 1) * 1 RTT.

c. Since the link is infinitely fast, we can assume that transfer time (the time to insert data into the link at the sender side) is effectively zero. Hence, we only need to be concerned with propagation delay of packet bundles, which is 0.5 RTT.

We need to send ceil((F / P) / 20) = 77 bundles of packets, the sender must wait 1 RTT between sending each bundle and an additional 0.5 RTT for the last bundle to propagate to the receiver. Factoring in the initial 2 RTT handshake, we arrive at

2.5 RTT + [ceil((F / P) / 20) - 1] * 1 RTT
= 6.28 seconds

This assumes that after the initial handshake, the sender sends a bundle, *then* waits 1 RTT. An alternative reading of this problem could be that the sende waits 1 RTT, *then* sends a bundle. This imposes an additional RTT's worth of overhead, so we would also accept

2.5 RTT + [(ceil((F/P) / 20)] * 1 RTT
= 6.36 seconds

d. Reasoning is the same as in c., except we have to send ceil(lg(F / P)) bundles of packets due to the exponential increase in bundle size. Total time can be expressed as

2.5 RTT + [(ceil(lg(F/P)) - 1) * 1 RTT]
= 1 second

By similar reasoning as above, we would also accept

2.5 RTT + [(ceil(lg(F / P))) * 1 RTT]
= 1.08 seconds

2.3) Solving this was essentially doing a lookup in the 4B/5B translation table for each nibble of the input bitstream.

The resulting translated bit string is:
11011 11100 10110 11011 10111 11100 11100 11101

2.5) Stuffed bits are indicated in bold. There were three of them.

110101111**1**0**0**...

We accepted answers both with and without sentinel frames, since it wasn't clear whether the question required them or not.

2.7) There was one error where seven consecutive ones are detected. There is an end-of-frame sequence at the end of the string. There is one stuffed bit to remove.

01101011111^10100111111**1(err)**0110**01111110(eof)**

Question 5)

There were many ways of making a convincing argument here. The main argument we expected here was that sending hundreds of gigabytes over the wide-area would take forever because of limited bandwidth and contention for links. In effect, for large data volumes the bandwidth of the "path" represented by putting hard drives on a truck (which is approximately the amount of data you can put on the truck / the amount of time the truck takes to get there, if you ignore data loading and unloading times) is much larger than the bottleneck bandwidth of any end-to-end path on the current Internet.

Grading here was a bit of a challenge, since there were several ways to make a convincing argument. If you're concerned about your grade on this problem, you can talk to us during office hours about it. The general rubric is as follows:

4 points total.
4 points: you gave us a convincing argument that was justified and displayed conceptual understanding.
3 points: the argument you gave us was pretty convincing, but was either not sufficiently justified or had some other minor problems.
2 points: we felt your explanation was based on a misunderstanding of how the networks we have studied so far in this course function, or you stated some points that were sort of reasonable but didn't justify them
1 point: unjustified one-sentence answers or those answers that we felt were based on a seriously flawed premise
0 points: you didn't write anything