Problem 1)

a) Application data, transport header, network header, and link header
b) Network layer, processes destination IP and forwards packets
c) When a packet moves across multiple hops, the link layer is removed first and thus when a frame reaches Host B the link layer will remove the link header first.

Problem 2) We know $R_s < R_c$

a) First packet last bit time: L/R_s, total time: $L/R_s + d_{prop} + L/R_c + d_{prop}$
Second packet FIRST bit sent time: L/R_s, LAST bit time: $2L/R_s$
Total time for 2nd packet: $2L/R_s + d_{prop} + L/R_c + d_{prop}$

$(2L/R_s + d_{prop} + L/R_c + d_{prop}) - (L/R_s + d_{prop} + L/R_c + d_{prop}) = L/R_s$

b) Yes if $R_s > R_c$ the 2nd link is slower and thus we run the risk of the 2nd packet not sending in time. The first packet is done transmitting at $L/R_s + d_{prop} + L/R_c$ and the 2nd packet arrives at the 2nd link at $T + L/R_s + d_{prop}$. If we do $L/R_s + d_{prop} + L/R_c <= T + L/R_s + d_{prop}$ we get $L/R_c <= T$

Problem 3)

a) Propagation delay = distance/seconds = $3.33 * 10^{-8}$ seconds
Transmission delay = length/rate
200 bit transmission delay = 200/200 = 1 second
100k bit transmission delay = 500 seconds

Non Persistent No Parallelism: (1 + 1 + 1 + 500) * 11 = 5533 seconds, each object requires a request and response packet. The first 3 are 200 bits long transmitted at 200 bits/second so we have 3 packets at 1 second each. The last one is 100k transmitted at 200 bits/second so it's 500 seconds. Since we got 11 objects we multiply the whole thing by 11.

Non Persistent Parallelism: $R_{parallel}$ is 200/10 = 20 bits/seconds. If we have 11 connections in parallel they each get 1/11th of the bandwidth (bandwidth is 200 bits/second). Receiving index files that are each 200 bits long at 200 bits/second is 1 second each, we have 3 index files so we get 3 seconds. The response is 100k bits long at 200 bits/second so we get 500 seconds. This sums to 503 seconds. For 10 packets in parallel, for each of them we got 2 connection packets and a request packet that are 200 bits long each so it would be 200/200 * 3 = 3 seconds. Since the $R_{parallel}$ is 20 bits/second for 100k bits we have 5000 seconds. The total would be 5003. 5003 + 503 = 5506 seconds.

Persistent No Parallelism: this would be 1 + 1 + 11 * (1 + 500) = 5513 seconds. We have 2 connection packets at the start and the request and response packet is 1 and 500 seconds respectively 11 times for 11 packets getting us 5513 seconds total.

Parallel non persistent HTTP doesn't have a significant advantage over a non parallel non persistent HTTP in this case. Additionally, persistent HTTP doesn't have a significant advantage over its non persistent case. The reason for the same-ish time between parallel non persistent and non parallel non persistent is due to the bottleneck from the rate of transmission. Since the largest object (100k bits) only travels at 200/bits/second the download is kinda stuck at 500 seconds.

Persistent HTTP doesn't provide a significant advantage over non persistent because persistence only eliminates the 2 connection packets (which take 1 second each) and that's nothing compared to the 500 second response packet time.

Problem 4)

a) /cs453/index.html, host is gaia.cs.umass.edu, so full URL is http://gaia.cs.umass.edu/cs453/index.html
b) HTTP/1.1
c) Persistent because we see Connection: keep-alive<cr><lf><cr><lf>
d) Content-Length: 3874, so 3874 bytes
e) <!doc, we know because we see <cr><lf><cr><lf> which separates headers from the actual body

Problem 5)

a) 142.250.217.142
b) Time to live response is 273, 2 minutes later it would be 273 - 120 (seconds) = 153
c) Only refreshes after 273 seconds, so it would only refresh after 19:00:00 + 273 seconds = 19:04:33.
d) The time to live response for NS records is 55416 seconds, meaning that it would be 1/25/2025 19:00:00 + 55416 = 1/26/25 10:23:36 (in the problem it says Jan of 2023 and 2025 so I will leave this note right here in case I get it wrong)

Problem 6)

a) Root to .edu to ucla.edu is 60 seconds each, so 3 * 60 = 180 ms. But we gotta add 5ms to the local DNS server RTT so we get 185ms.
b) ucla.edu is already in the cache so we only have to go to the bruinlearn server so it's 60 ms + 5 ms for local DNS time = 60ms.
c) The root is cached but however we have to visit .com and gradescope so those two would take us 120 ms + 5ms for RTT between local DNS so it's 125 ms.

d) Since .com is already cached, going to google takes 60 ms and RTT takes 5ms. So the total time is 65ms.