Problem 1)

a) Application data, transport header, network header, and link header
b) Link layer
c) When a packet moves across multiple hops, the link layer is removed first and thus when a frame reached 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) For Non-persistent HTTP, each object has their own handshake, get, and response, meaning that total time per object is multiplied by 11. On the other hand, for parallel, non-persistent HTTP, each object gets 1/10 of the bandwidth. While handshake and get still applies for each object, if we assume full parallelism we can download all 10 objects simultaneously. For persistent HTTP, we only have one handshake for all objects but the get request is still needed for each object. But response packet time is still the bottleneck, but it is a lot better than the parallel, non-persist case.

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

Problem 6)

a) Cache is empty, so time would be query from local to root DNS + response from root DNS = 60 + 60 = 120 ms. Query from local DNS + response from .edu server = 60 + 60 = 120 ms. Query from local DNS to ucla.edu + response from ucla.edu = 60 + 60 = 120 ms. Final response from your computer to local DNS + local DNS back to computer = 5 + 5 = 10ms. Sum is 370 ms.

b) You only have to query local DNS since we are visiting bruinlearn.ucla.edu, not just ucla.edu. We also gotta include final response time. So 120 + 10 = 130 ms. TTL is 5 hours so if we query one minute the time is irrelevant.

c) Gradescope is a new domain name so we gotta do it all again. It would be the same as part a which is 370 ms.

d) TTL is 5 hours, so we don't have to do the root query. However since google is completely different we gotta do everything other than root query again so it's 370 - 120 = 250 ms.