1. The application layer and transport layer only operate on the hosts at each end for a given application, these layers rely on the network layer and below to get segments across the network core which forwards network layer traffic across the network without operating on or inspecting the contents of the higher layers protocol's data.
2. Persistent HTTP keeps all HTTP requests queued in one TCP connection but only spends one RTT to open one connection however each request happens in series. Non-persistent HTTP restricts each connection to one request thus each request has the additional overhead of having to wait 1 RTT to open a TCP connection before the request can be made however multiple TCP connections can be made in parallel meaning no request blocks any other request. Additionally, HTTP operates over TCP and TCP fairness operates such that each connection effectively gets an equal division of bandwidth on a given link, more TCP connections means a larger chunk of bandwidth for that application's traffic.
3. Here is what we know:
    - Link delay = Nodal delay + Queue delay + Transmission delay + Propagation delay
    - There are two links between the client and the server, each link is identical in this case
    - Link speed: 1 Mbps, Link length: 1000 km, Link propagation speed: 250000 km/s
    - Propagation delay = length over speed
      1000 km / 250000 km/s = 0.004 s = 4ms
      (if you try to make the units work and cancel out it makes sense)
    - Transmission delay = packet length / speed =
      N bytes / 1000000 bits/s = N bytes / 125000 bytes/s = N bytes / 125 bytes/ms = N/125 ms
    - Nodal delay = 0, Queue delay = 0 as stated in the question

    Delay to get one packet of length N over one link: (0 + 0 + N/125 + 4) ms = (N/125 + 4) ms

    - Packet length for N bytes of data: 125 + N bytes, N = 0 for opening and confirming data packet size for opening and confirming data: 125 + 0 = 125 bytes
    - Request is 250 bytes: packet size: 125 + 250 = 375 bytes
    - Response is 500 bytes for the header, 4000 bytes for the body:
      packet size: 125 + (500 + 4000) = 4625 bytes
    - All these calculations are less than the MTU of 8192 so no none of the data needs to be broken up into multiple packets

    We need to look at the packets that are moving across the network:

    125 bytes → to server to opening the connection
    125 bytes ← to client confirming the connection is open
    375 bytes → to server, HTTP request
    4625 bytes ← to client, HTTP response

    (This is a simplified case, if you added additional packets for a proper three-way handshake in a TCP connection, your answer would also be correct)

    Two links means we apply the formula derived above for each link

    (N/125 + 4) + (N/125 + 4) in ms

→ (125/125 + 4) + (125/125 + 4) = (1+4) + (1+4) = 10

← (125/125 + 4) + (125/125 + 4) = (1+4) + (1+4) = 10

→ (375/125 + 4) + (375/125 + 4) = (3+4) + (3+4) = 14

← (4625/125 + 4) + (4625/125 + 4) = (37+4) + (37+4) = 82

None of these communications can be pipelined so add them together as is.

a. Only consider the two packets to open the connection:
   10ms + 10ms = 20ms
b. Consider the whole communication:
   10ms + 10ms + 14ms + 82ms = 116ms

4. HTTP 1.1 is defined in such a way that the client must always make a request, a server cannot push data to a client using HTTP alone. A web client can opt to poll a server periodically for updates to give the impression updates are happening in real time or the application must switch to another protocol.

   As a side note, the web sockets protocol can be used for this purpose but outside the scope of this course. Webhooks are another topic outside the scope of this course where a client and a server switch their role such that a server acts as a client and connects to a webserver hosted on host that the application client is running on in order to make requests to the client.
5. Here is what we know:
   a. 1Gbps link
   b. 50 requests / s
   c. 2MB / request
   d. We can calculate data rate needed:
      50 requests/s * 2MB/requests = 100MB / s = 800Mbps
   a. 800Mbps / 1Gbps = 800Mbps / 1000Mbps = 0.8 = 80%
   b. Half the requests, adjust the numbers from above
      (50/2) requests/s * 2MB/requests = 50MB / s = 400Mbps
      400Mbps / 1Gbps = 400Mbps / 1000Mbps = 0.4 = 40%
   c. This would not provide any advantages, the extra connection through the proxy server will increase the delay to get the content since an extra hop has been added between the client and the server.
6. .
   a. SMTP can be configured to relay outbound mail to another SMTP server before reaching its destination, additionally the DNS MX record for a domain may point to a specific SMTP server but that server may not be the server that hosts the mailboxes, it may forward mail it receives toward an internal SMTP server to handle functions such as spam detection and filtering or to pass it along to the SMTP server that also hosts the mailboxes.
   b. It is possible that a web server accepts RFC2822 messages as the body of a POST request and uses that payload to forward the message over SMTP as its function. This is important as web browsers do not support connecting over SMTP but can use HTTP and can use it as a relay using web technologies such as AJAX to build a web client.

7. RFC 2822 is the standard that defines the format of an email message. SMTP is designed to deliver RFC 2822 messages. SMTP was designed when multiple character encodings were prevalent across the internet and so defined ASCII as the protocol's character encoding making all non-ASCII byte values invalid. RFC 2822 defines a few methods for encoding data that is not ASCII into ASCII data, an example of this is base-64 which encodes groups of three bytes into groups of four ASCII characters. This allows binary data such as images to be embedded in an RFC 2822 message while keeping the message fully ASCII. Since each character is a byte in ASCII, this inflates the binary data to be 4/3 of its original size thus taking up more space than is necessary but while maintaining support for mail systems that do not support non-ASCII characters.
8. The two main attacks against DNS are DDoS attacks and cache Poisoning
    a. DDoS attacks attempt to flood root and TLD servers with bogus traffic to force legitimate requests to be dropped. Packet filtering by those servers mitigate the attack as well as having long lived cache lifetimes so that should an attack succeed, local DNS servers would not be immediately affected as they will still be using a cached value.
    b. Cache poisoning is another attack, where an attacker tries to forge a response from the server. DS records mitigate this at a design level by cryptographically signing DNS records and the need to be able to intercept traffic mitigates this at a practical level.
9. .
    a. A WHOIS database is used to map DNS domain names to owners and registrars who register domains on behalf of domain owns as well as the authoritative DNS server for a domain
    b. CIRA's WHOIS service gives the following nameservers for
        i. concordia.ca
            1. Name Server: ns-a.concordia.ca
            2. Name Server: ns-b.concordia.ca
            3. Name Server: ns1.cc.umanitoba.ca
            4. Name Server: ns1.zonerisq.ca
            5. Name Server: ns2.zonerisq.ca
        ii. montreal.ca
            1. Name Server: coby.ns.cloudflare.com
            2. Name Server: tani.ns.cloudflare.com
    c. Omitted for length, findings:
        i. Local DNS server 132.205.96.93 returns results for all domains
        ii. ns-a.concordia.ca only returns results for domains under concordia.ca
        iii. coby.ns.cloudflare.com only returns results for domains under montreal.ca
        iv. NS records match the nameserver records from the WHOIS database
        v. A records refer to IP addresses
        vi. MX records are other domain names which have their own A records to the SMTP server for that domain
    d. concordia.ca only has one A record, montreal.ca has two A records
    e. Concordia has the IP range 132.205.0.0 to 132.205.255.255 or simply 132.205.0.0/16

    f. You can determine a lot of information about a site using WHOIS and nslookup such as the organization in charge of network management from the WHOIS record to service providers from looking up the IP range of an A record under that domain.
    g. WHOIS databases need to be public record in the same way land ownership is public record so that ownership and responsibility can be known.
10. P2P is mainly beneficial at large scale. Client server file distribution time is linear, as the number of hosts grow, the distribution time grows with it. P2P file distribution time starts off linear as but becomes constant asymptotically for large numbers of hosts. For a small number of hosts, the distribution time remains in the linear zone as it takes time to distribute the file.
11. Tit-for-tat is a P2P strategy where peers prioritize uploading to peers which have let them download from them. This causes a problem for new peers which have no content initially so no other peers can download from them. This means new peers keep an initially low priority for a relatively long time until they have enough content to share and peers which have content have a high download priority despite having content already. This strategy is not beneficial to new peers but is beneficial to the swarm as a whole as it causes new peers to remain in the swarm long enough to share content in order the complete their download mitigating the issue of peers greedily joining a swarm to download without uploading back to the swarm.
12. The available bandwidth was reduced either by network congestion, degradation of link quality or switching to another ISP (such as going from Wi-Fi to LTE). The video is streamed via DASH or HLS which provided the client with a file which contains a list of URLs to short segments of video in different qualities. When the software detected a drop in available bandwidth, it switched to a lower quality set of video segments while a previous segment was playing.