Problem 1 (20 points)

1.1 Circle zero or several application-layer protocols that use only TCP as their transport layer protocols?

(a) HTTP 1.1/2	**(c) SMTP**	**(e) BitTorrent**	**(g) MPEG/DASH**
(b) QUIC	**(d) IMAP/POP3**	(f) DNS	(h) Skype/VoIP

1.2 Circle zero or several application-layer protocols that are stateful?

(a) HTTP 1.1/2	**(c) SMTP**	**(e) BitTorrent**	(g) MPEG/DASH
(b) QUIC	**(d) IMAP/POP3**	(f) DNS	**(h) Skype/VoIP**

1.3 Circle zero or several statements that are **TRUE** for a peer-to-peer system?

(a) All systems always need to be on
(b) Transferring a file is faster than an equivalent client-server architecture
(c) They are not as scalable as client server architecture
(d) Are easier to implement than client-server systems

1.4 Circle zero or several pieces of information one **CANNOT** get by looking at an HTTP request message?

(a) Name of the web-page	(c) Server's port number	**(e) Requester's IP address**
(b) Server's host name	**(d) Server's IP address**	**(f) Full URL of the request**

1.5 (10 points) Fill in the blanks:

The network protocols (and protocols in general) define

_____**message format,** _____,

___ **communication sequence (order of messages sent)**_____, and _____ **actions to take** _____.

The most common HTTP method types are

_____**GET**_____ and _____**POST**_____.

DNS protocol is a highly available database because DNS zone information (resource records) can be

_____**replicated**_____ and _____**cached**_____.

HTTP protocol can scale because WEB data can be

_____**replicated**_____ and _____**cached**_____.

The common function (at least one) between TCP and UDP transport-layer protocol is

_____**multiplexing / demultiplexing**_____.

In addition to this function, TCP also provides (at least 2 for full credit)

____**delivery guarantees**__, __**flow control**_____ and ___**congestion control**_____.

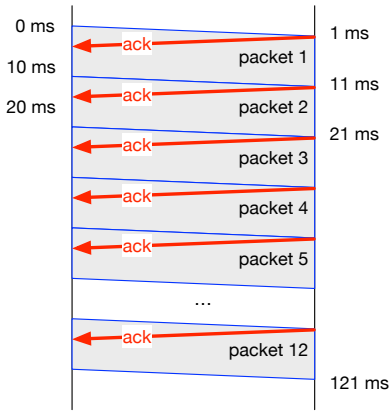
Problem 2 (20 points) Two hosts A and B are connected by a link with bandwidth of 1 Mbps (10^6 bits-per-second) and propagation delay between A and B is 1 millisecond. Host A has a 500,000-bit file to send to host B. A uses GoBackN reliable transport protocol and divides the file into 10,000-bit packets. The GoBackN protocol uses a fixed window size of 4 packets. You may assume the *transmission time* of ACK packets is negligible and no data or ACK packet ever gets lost.

2.1 (6 points) How long will it take before the 12th packet has completely arrived at Host B? (drawing a diagram may help answer this question).

121ms

To put 12 packets on the wire, would take at least 12 * 10,000 bits / 1,000,000 bits/s = 120ms After a single packet is put on the wire, it takes 2*propagation delay to get an ACK back = 2ms. Effectively host A will receive an ACK for the first packet while sending the second packet, which means there is only 1 full packet that is waiting ACK at any given time and another packet in progress.

Therefore, for host B to get all 12, would need transmission time for all packets and one way propagation delay



2.2 (2 points) How long will it take before the entire file is received by Host B?

50 * 10ms + 1ms = 501ms

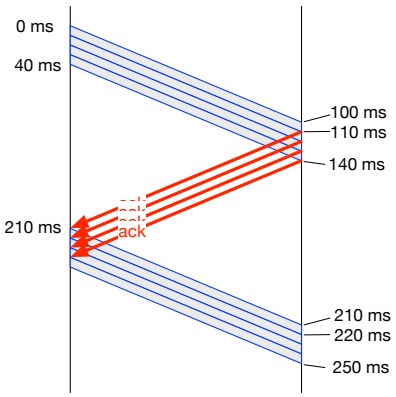
2.3 (6 points) How long will it take before the entire file is received by Host B if propagation delay is increased to 100 milliseconds?

There is still 50 * 10ms of transmission delay and 100ms of one way propagation delay

In addition, after 4 packets, there is a gap of 170ms waiting for an ACK for the first packet to arrive before sending packet #5.

For 50 packets and N=4, we have 13 groups of packets (12 full and 1 half) with 12 gaps in the middle:

Total delay: 50*10ms + 100ms + 12*170ms = 2,640 ms (2.64 seconds)

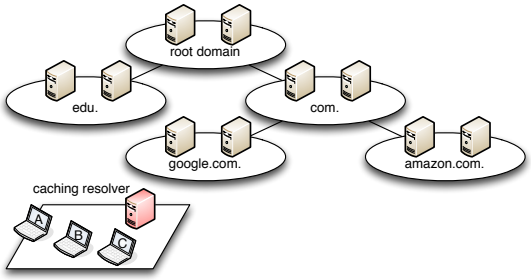


2.4 (6 points) Assuming propagation delay stays 100 milliseconds, is there a way for the file to be delivered to the host B faster by adjusting the window size? If so, what is the minimal window size that would allow the file to be received at B with shortest possible time (assume no other settings are changed)?

To send efficiently, need to avoid the gap, so N ~= 1 + RTT/transmission delay = 1 + 200ms / 10ms = 21 packets

Min time to transfer the whole time (no transmission gaps): 50*10ms + 100ms = 600ms

Problem 3 (20 points) Consider the following environment with a local DNS caching resolver and a set of authoritative DNS name servers.



Assume that initially,

- the caching resolver cache is **empty**,
- TTL values for all records is **1 hour**,
- RTT between stub resolvers (hosts A, B, and C) and the caching resolver is **20 ms**,
- RTT between the caching resolver and any of the authoritative name servers is **150 ms**
- There are **no packet losses**
- All processing delays are **0 ms**

3.1 (4 points) At T=0 min, Host-A sends a query for "A record for amazon.com", and after receiving the answer sends a query for "A record for www.amazon.com". How long did it take to receive all the answers?

First = 20ms (caching) + 150 (query to root) + 150 (query to .com) + 150 (query to amazon.com)

Second = 20ms (caching) + 150 (query to amazon)

Total 640 ms

3.2 (3 points) At T=40 min, Host-B sends a query for "MX record for google.com" that returns

```
google.com.          3600   IN     MX     10 primary.google.com.
google.com.          3600   IN     MX     30 backup.google.com.
primary.google.com.  3600   IN     A      74.125.28.27
backup.google.com.   3600   IN     A      173.194.211.27
```

(Similar to NS records, the DNS server may return "glue" A/AAAA records in addition to the requested MX records.) How long did it take to get the answer?

At t=40 no caches yet expired, so 20ms (caching) + 150 (query to .com) + 150 (query to google.com) = 320ms

3.3 (5 points) At T=70 min, Host-C sends a query for "AAAA (IPv6) record for mail.google.com", following at T=75 mins with a query for "AAAA (IPv6) record for hangout.google.com". How long did it take for Host-C to receive each of the answers (i.e., relative to T=70min for the first, and relative to T=75 mins for the second)?

Info about .com has expired at this point, but info about google.com NS servers is still in cache

First = 20ms (caching) + 150 (google.com)
Second = 20ms (caching) + 150 (google.com)

3.4 (5 points) List DNS records that the caching resolver has at T=90 minutes

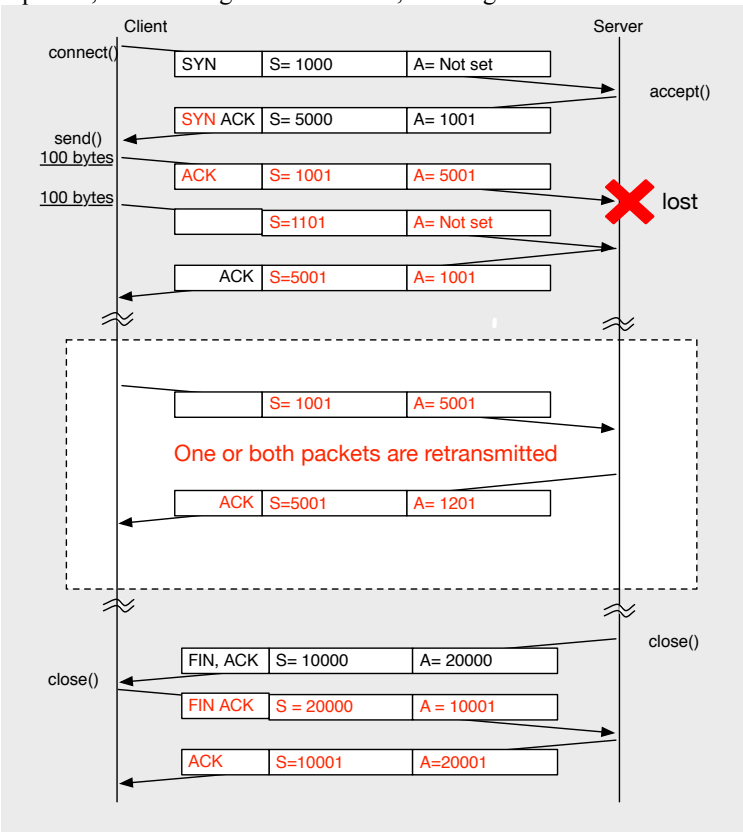
All returned in 3.2 + all returned in 3.3:

Google.com/MX, primary.google.com/A, backup.google.com/A, google.com/NS (10 mins remaining)
Mail.google.com/AAAA (40 mins remains)
Hangout.google.com/AAAA (45 mins remains)

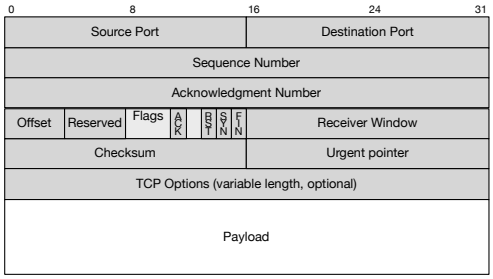
3.5 (3 points) At T=100 minutes, all the authoritative servers for **.com** go offline. Circle the domain names that can be resolved by Host-A? Given the ambiguity, full credit for this question for any answer

```
(a) www.google.com     (b) hangout.google.com   (c) doc.google.com
(d) www.amazon.com     (e) video.amazon.com     (f) aws.amazon.com
```

Problem 4 (20 points) The following diagram shows a sequence of TCP packets for a client/server from your project 1, which include some of the sequence, acknowledgement numbers, and flags.



4.1 (10 points) In the figure above, fill in all the missing values for sequence, acknowledgement numbers and flags (SYN, ACK, FIN). For acknowledgement number write "Not set" if acknowledgment flag not set.



4.2 (5 points) One of the packets got lost. In the dotted box above, add the missing exchanges between the client and the server just after the loss has been detected. In the exchange, include flags, sequence number, and acknowledgement number (if applicable).

4.3 (5 points) What is the theoretical maximum of the TCP pipeline?
For a link with 500ms round trip delay imaginary 1000 Tbits/s link bandwidth, what is the maximum throughput that TCP protocol can sustain for that link (assume maximum packet size is 1000 bytes, hosts have infinite amount of buffering memory)?

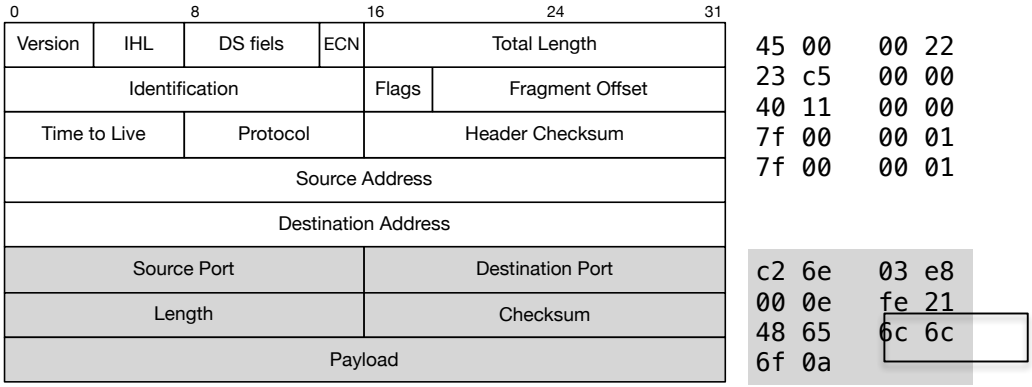
To fill the pipe, TCP would need window N (in packets) = (1 + RTT/transmission delay) (from probem 2)
Transmission delay for one packet = 1000 bytes * 8 bits/byte / 10^12 bits/s ~= 10^-8 seconds

N = (1 + 0.5 / (10^-8)) = 1 + 5 * 10^8 ~= 5 * 10^8 (packets). In bytes, Nbytes = 1000 * 5 * 10^8 = 5 * 10^12 ~= 2^44

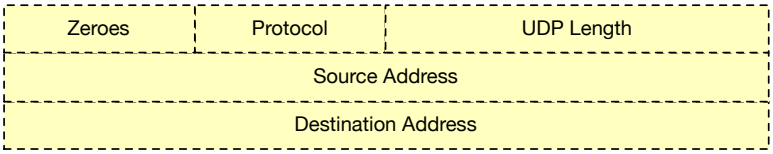
TCP can only have at most 2^31 window bytes (set of sequence numbers divided by 2). Therefore, maximum what possible to achieve for this link:

2^31 bytes (maximum window size) / 0.5 second ~= 4294967296 bytes/s = 34359738368 bits/s ~= 32 Gbits/s

Problem 5 (20 points): The following diagram shows the UDP packet header and HEX value of one of the captured UDP packets.



Pseudo-header format for UDP/IPv4



5.1 (5 points) Check correctness of UDP checksum. If it is incorrect, what should be the correct checksum?

Checksum in the packet: 0xFE21 and it is not correct.

Actual checksum = 1-complement of sum(pseudo header, UDP)
Pseudo header= 0xFE21; UDP = 0x1EA3F = 0xEA40
checksum = 1-compl of 1-complement of 0x1E861 = 1-complement of 0xE862 = 0x179D.

5.2 (5 points) Check correctness of IPv4 checksum. If it is incorrect, what should be the correct checksum?

Checkum in the packet 0x0000

Actual: 0x176FA = 0x76FB = 0x8904

5.3 (6 points) Please describe how this packet can be delivered to the destination application (i.e., how OS de-multiplex this packet) and on which port number this application should be listening on.

IP to demultiplex to host, protocol to determine which protocol table to lookup, (it is UDP packet), use just destinationIP/destinationPort to find an entry in that table, if there is an associated socket, deliver packet to that socket/application.

5.4 (4 points) Finish the following statements about UDP protocol:

The maximum size of a UDP payload is ____**2^16**_____

The range for UDP port numbers is _____**0-2^16**_____

For a computer with two IP addresses (e.g., one for wireless and one for wired), there could be ____**2*2^16**_____ maximum number of distinct UDP server applications.

To prevent anybody else to start a UDP server application, one need to start at least ____**2^16 (binding on 0.0.0.0)**_____ number of applications, each creating one socket, binding, and listening on a single port.

Problem 6 (20 points)

6.1 (4 points) Assume that you want to send a secret message over email using PGP/GPG to a person you just googled on the Internet (you found his email and have a secret question to ask). Will you be able to do that? If yes, how (conceptually), if no, why?

No, we need the person's public key to communicate secretly.

Keybase.io give an approximation for that (through it you can check that the person that owns private key has control over facebook, github, linkedin, etc. accounts). The key lookup services that are used by default in GPG are **not** secure at all, as anybody can publish a key pretending to be anyone).

6.2 (4 points) Let's say you sent an email to the professor. If you haven't used PGP/GPG, he will not be able to know for sure that it came from you. List **at least two** facts that he can learnt from the received email that the sender couldn't fake. What can you do to ensure that the email is from you, including any **out-of-band** process that may be needed.

Objective facts:

- Last few SMTP servers used to deliver message (because the professor trust its mail service/Gmail to some extent)
- Timestamp it was received by one the last SMTP servers (e.g., gmail server that is final destination of the message)
- DKIM/SPF verification done by last SMTP servers

What can you do to ensure email is from you? Sign it, e.g. using PGP, and make sure to tell (out-of-band) your public key to the professor (either in person, over the phone, send in U.S. mail, etc.)

6.3 (4 points) Let's say you go to a website over HTTPS protocol and get a warning that something is wrong with the certificate and browser rejects to proceed. List **at least four** reasons what can be wrong with a brief explanation what could have happened.

- Certificate expired (lifetime for which the certificate was issues is over)
- Certificate revoked (due to name change or privacy compromise)
- Untrusted Certificate (not issued by a trusted authority or self-signed)
- Invalid Certificate (certificate might be valid for www.example.com but not example.com)
- Out of date browsers
- System time is not real time
- Website might be using outdated SHA-1 algorithm
- File that stores the certificate might have become corrupted (cert8.db)

6.4 (4 points) Your professor travels a lot and whenever he has a chance he issues a DNS query for "A record for youtube.com". So far, he collected quite a bit of a collection of different responses. List **at least four** reasons why he gets different responses.

- Geographical proximity may decide the IP address
- Load balancing may decide the IP address
- Local caching resolvers might be changing
- Change in IP address of youtube.com or Multiple IP address of youtube.com
- Captive Wifi (Fake wifi in every hotel)

6.5 (4 points) HTTP/2 supports multiple streams and proactive push of data by the server. Give at least 2 reasons why people decided to develop QUIC.

- Flexible congestion control
- Solves Head-of-line blocking
- Faster communication / Reduced latency / Connection ID reuse
- Forward error correction