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

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

(a) HTTP 1.1/2
(b) QUIC
(c) SMTP
(d) IMAP/POP3
(e) BitTorrent
(f) DNS
(g) MPEG/DASH
(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
(b) Server's host name
(c) Server's port number
(d) Server's IP address
(e) Requester's IP address
(f) Full URL of the request

1.5 Fill in the blanks:

The network protocols (and protocols in general) define

logical communication between hosts

_____, and _____.

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

cached and delegated.

HTTP protocol can scale because WEB data can be

cached and replicated/distributed.

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

multiplexing, demultiplexing.

In addition to this function, TCP also provides

reliable, in-order delivery, flow control and congestion control.

50

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).

first packet

$0.001 \cdot 2 + \frac{10000}{10^6} = 0.012\,s$

round trip add $\frac{10000}{10^6} = 0.01\,s$ per packet

12th $0.012\,s + 0.01\,s \cdot 11 - 0.001\,s = 0.121\,s$

only arrive

2.2 (6 points) How long will it take before the entire file is received by Host B?

first packet

$0.001 \cdot 2 + \frac{10000}{10^6} = 0.012\,s$

50 packets

$0.012\,s + 0.01\,s \cdot 49 = 0.502\,s$

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?

first packet

$0.1 \cdot 2 + \frac{10000}{10^6} = 0.21\,s$

50 packets

$0.21\,s + 0.01\,s \cdot 49 = 0.60\,s$

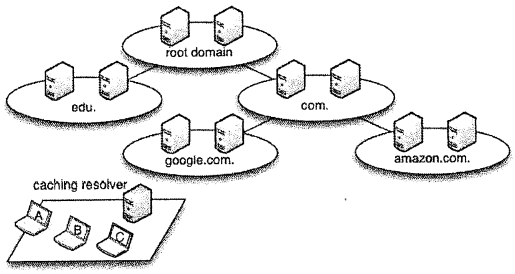
2.4 (8 points) Assuming propagation delay stays 100 milliseconds, is there a way for the file to be delivered to the host B faster by adjusting the window size? If so, what is the minimal window size that would allow the file to be received at B with shortest possible time (assume no other settings are changed)?

window size $N = 10$

propagation $\frac{0.1\,s}{0.01\,s} = 10$

bandwidth

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?

for amazon.com
150ms get com from root
150ms get NS of amazon.com from com
150ms get amazon.com from NS of amazon.com
20ms stub ⇔ caching

for www.amazon.com
assuming www.amazon.com is not delegated to www zone
150ms get www.amazon.com from NS amazon
20ms stub ⇔ caching

total 640ms

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

```
google.com.          1h 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?

150ms get NS of google.com from com
150ms get MX of google.com from NS of google.com
20ms stub ⇔ caching

total 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)?

Assume mail.google.com is in its mail zone, and hangout.google.com is in its hangout zone.
150ms get NS of mail.google.com from NS of google.com
150ms get AAAA mail.google.com from NS
20ms stub ⇔ caching

320ms

hangout.google.com is equivalent 320ms

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

google.com NS
google.com MX primary
google.com MX backup
primary.google.com A
backup.google.com A

mail.google.com NS
ns.mail.google.com A
mail.google.com A
hangout.google.com NS
ns.hangout.google.com A

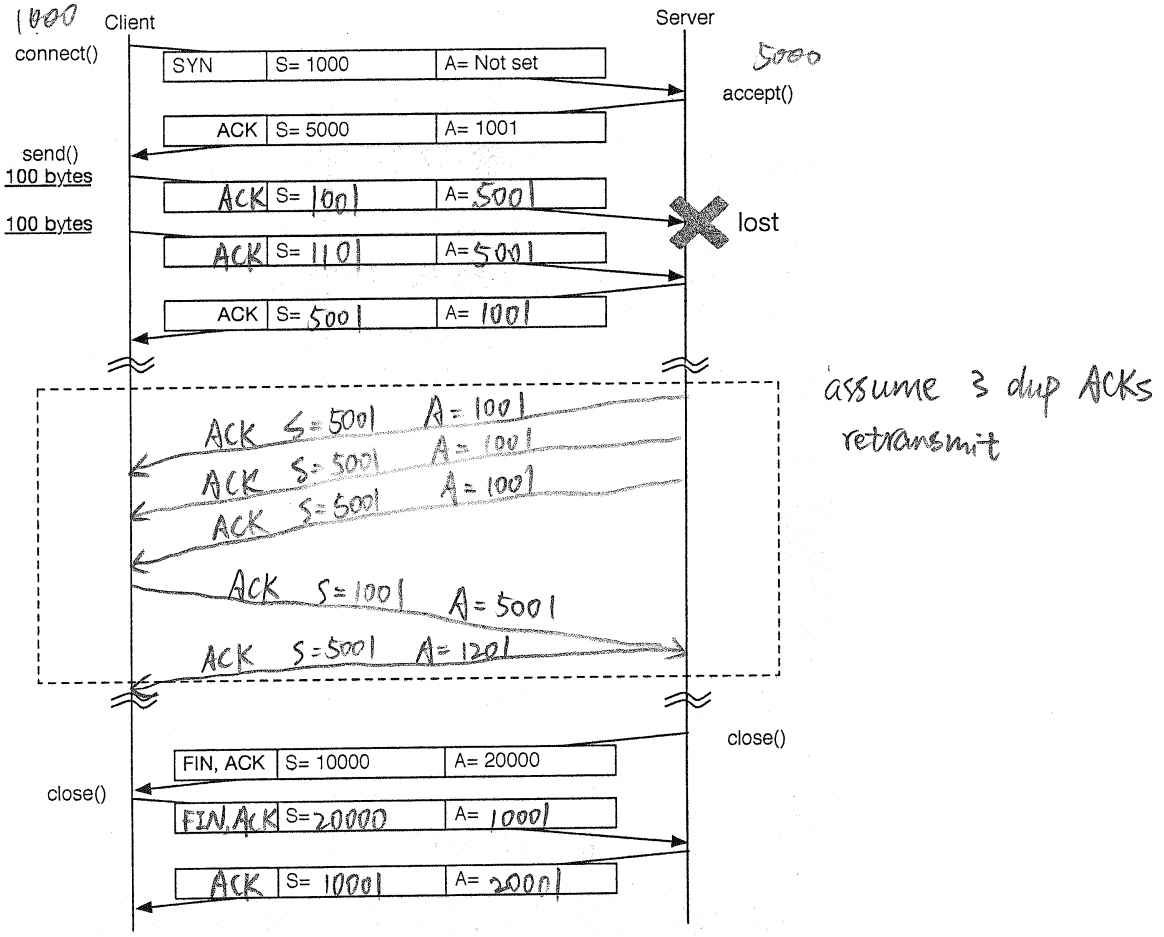
only get NS for hangout.google.com

hangout.google.com A

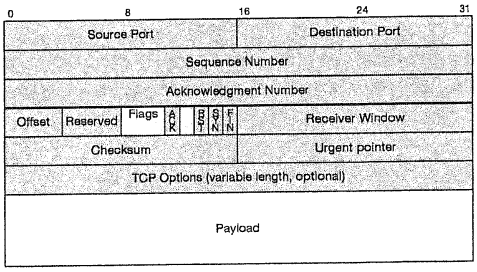
3.5 (3 points) At T=100 minutes, all the authoritative servers for **.com** go offline. Circle the domain names that can be resolved by Host-A?

(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)?

connection establishment

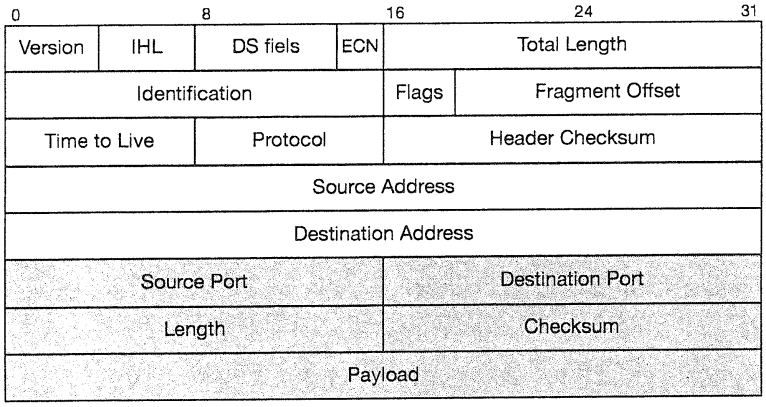
1RTT 500ms

left 500ms = 0.5s

1000Tb/s · 0.5s = 500Tb → 5×10^8 packets/s

10^9

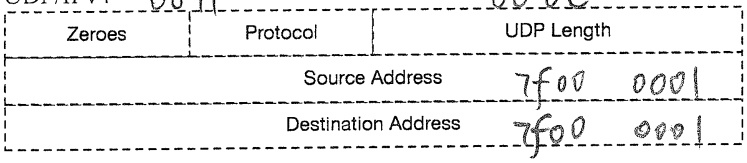
Problem 5 (20 points): The following diagram shows the UDP packet header and HEX value of one of the captured UDP packets.




```
45 00   00 22
23 c5   00 00
40 11   00 00
7f 00   00 01
7f 00   00 01

c2 6e   03 e8
00 0e   fe 21
48 65   6c 6c
6f 0a   6b 6f
```

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?

in pseudo header + UDP (checksum 0) (one's complement sum)

0011 + 000e + 7f00 + 0001 + 7f00 + 0001 +
c26e + 03ef + 000e + 4865 + 6c6c + 6f0a = e962

one's complement of e962

169d

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

IP packet with checksum 0 (one's complement sum)

4500 + 0022 + 23c5 + 0000 + 4011 +
7f00 + 0001 + 7f00 + 0001 + c26e + 03ef +
000e + fe21 + 4865 + 6c6c + 6f0a =

take one's complement of that sum

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.

OS deliver packet based on scr IP, dst IP, src port, dst port.
This packet is delivered to port 03e8 (hex), OS checks dst IP & port, deliver this packet to applications listening on port 0x03ef (hex)

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

The maximum size of a UDP payload is 66535 ($2^{16}-1$)

The range for UDP port numbers is 0 ~ 66635

For a computer with two IP addresses (e.g., one for wireless and one for wired), there could be 133072 maximum number of distinct UDP server applications. 66536·2

To prevent anybody else to start a UDP server application, one need to start at least 66536 number of applications, each creating one socket, binding, and listening on a single port.

256 × 256 = 65536

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?

Yes, assuming he has also posted his public PGP key and is authentic. You will encrypt the message with his public key, and send this encrypted message to him through email. Only he will be able to read the message

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.

Assuming that you have published your public key and you and your key is trustworthy. You encrypt the message with your private key, and professor can decrypt through your public key

① Only you can encrypt message with your private key ② professor with your public key can decrypt

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.

① expired: the certificate is out of date, needs to be renewed
② wrong host: the certificate is valid, but is for another host
③ untrusted root: the authority that signed this certificate cannot be trusted
④ revoked: the certificate is revoked, or no longer valid

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.

youtube.com is distributed through CDN across the globe. DNS servers for CDN resolves IP closest to client, which can serve clients faster. CDN pushes popular content shared by users to geographically distributed CDN nodes IPs closer to clients have lower latency and they are based on geolocation. This distributed system allows higher throughput and better reliability.

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.

QUIC uses UDP, which allows out-of-order delivery, and therefore no head-of-line blocking.
QUIC allows connection ID reuse, so no need for connection establishment if visited this host before.