CS118 Midterm

Karen Li

TOTAL POINTS

106 / 120

QUESTION 1

1 20 pts

1.1 1.1 (TCP as a transport) 3 / 3

- **- 0 correct**
- - 0.5 Incorrectly select
- - 0.5 Incorrectly select
- - 0.5 Incorrectly select
- - 1 should select 3 correct options
- - 1 should select 3 options
- - 1 should select 3 options

1.2 1.2 (stateful) 2.5 / 3

- - 0 Correct
- **- 0.5 Select a wrong answer**
- - 0.5 Select a wrong answer
- - 0.5 Select a wrong answer
- - 3 Wrong answer, leave blank

1.3 1.3 (p2p) 1 / 1

- **- 0 Correct**
- - 1 Incorrect (b is not chosen)
- - 0.5 At least one other (incorrect) options selected

1.4 1.4 (HTTP request/response info) 2 / 3

- - 0 Correct
- - 1 missing d
- - 1 missing e
- **- 0 missing f**
- **- 1 One wrong answer (a-c)**
- - 2 Two or more wrong answers (a-c)

1.5 Protocol is ... 2 / 2

- - 1 At least one correct (format/order/actions)
- **- 0 Two or more correct (format/order/actions)**
- - 2 Incorrect

1.6 Most common HTTP method 2 / 2

- **+ 1 GET**
- **+ 1 POST**
- + 0 Incorrect/Blank

1.7 DNS and HTTP 1.5 / 2

- **+ 0.5 replicated**
- **+ 0.5 cached**
- + 0.5 replicated
- **+ 0.5 cached**
- + 0 Incorrect

1.8 Common TCP/UDP functions, unique TCP functions 4 / 4

- **+ 2 multiplexing/demultiplexing/error detection**
- + 1 half credit to common function
- **+ 1 delivery guarantee**
- **+ 1 flow control**
- **+ 1 congestion control**
- + 0 Incorrect
- + 1 Incorrect: send/connect/accept/listen calls

QUESTION 2

2 20 pts

2.1 2.1 Delay for 12th packet 3 / 6

- + 6 Correct
- **+ 3 Correct transmission delay**
- - 1 Incorrect number of RTTs
- + 0 Incorrect

2.2 2.2 Total delay 2 / 2

- **+ 2 Correct 501ms, accepted answer if +-100ms**
- + 0.75 Correct transmission delay
- + 0.75 Correct total number of packets
- + 0 Math problem
- + 0 Incorrect

2.3 2.3 Total delay with 100ms propagation delay 6 / 6

- **- 0 Correct (correct 2.64s, accepted 2-3s)**
- - 1 No or incorrect explanation provided

- 3 Wrong value, but within reasonable range from the correct (1-2, 3-5)
- 0.5 Incorrect RTT calculation
- 6 Incorrect

2.4 2.4 Shortest delay / adjust window 3 / 6

- 0 Correct (20-21, delay ~600ms)
- 0.5 No (or incorrect) optimal delay calculated
- 1.5 Mentioned value >21, but no or incorrect explanation provided
- 6 Incorrect
- **3 Mentioned to increase window, but <21 or way too many**

QUESTION 3

3 20 pts

3.1 3.1 Query for amazon.com/A 4 / 4

- **0 Correct**
- 2 Second query incorrect
- 1 Problem with one of the queries
- 4 Incorrect / no answer

3.2 3.2 Query for google.com/MX 3 / 3

- **0 Correct**
- 1 Issue with the answer (one unnecessary query)
- 1.5 Didn't include query t google.com NS
- 3 Incorrect / missing / more than one unnecessary query

3.3 3.3 Query for mail/hangout.google.com/AAAA 5 / 5

- **0 Correct**
- 1 Extra (or missing) query for 1st query
- 1 Extra (or missing) query for 2nd query
- 2 No more than two unnecessary queries for one of the queries
- 3 More than 2 unnecessary queries for one
- 5 Incorrect / missing

3.4 3.4 List cached records 4 / 5

+ **2 google.com/MX, primary.google.com/A, backup.google.com/A (from 3.2)**
+ 1 google.com/NS
+ **2 mail.google.com/AAAA, hangout.google.com/AAAA**

- 0.75 One wrong domain
- 1.5 Two wrong domains
- 1 Type of records not specified

+ 0 Incorrect / missing

3.5 3.5 Reachable 3 / 3

- **0 Correct**

QUESTION 4

4 20 pts

4.1 4.1 Sequence numbers 10 / 10

+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ **1 One correct sequence number of flag**
+ 0 Incorrect/blank

4.2 4.2 Missing exchange 5 / 5

- **0 Correct**
- 0.5 One or more incorrect / unnecessary / missing exchanges
- 0.75 One flag or sequence number is wrong
- 1.5 More than one flag or sequence number is wrong
- 2.5 Sequence numbers / flags (or their relation) not shown
- 5 Incorrect / missing

4.3 4.3 Max TCP pipeline 3 / 5

- 0 Correct (or close)
- **2 Too large or too small**
- 2 Incorrect statement that there is no limit
- 2 The result is not throughput (bit/s or byte/s)
- 0.5 Right direction, but didn't give complete answer
- 5 No attempt
- 3 Attempted, but didn't give the answer

QUESTION 5

5 20 pts

5.1 5.1 UDP checksum 5 / 5

+ **1.5 Found checksum in the packet**
+ **4 Attempted to calculate 1-complement of 1-complement sum of 16-bits**
+ 1 Gave an answer without basis
+ 2.5 Calculated sum, but didn't indicate 1-complement of 1-complement / not correct items added
+ 2.5 Attempted calculate but not 1-complement of 1-complement sum or not 16-bit numbers
+ 0 No attempt / incorrect

5.2 5.2 IPv4 header checksum 5 / 5

+ **1.5 Found checksum in the packet**
+ **4 Attempted to calculate 1-complement of 1-complement sum of 16-bits**
+ 1 Gave an answer without basis
+ 2.5 Calculated sum, but didn't indicate 1-complement of 1-complement / not correct items added
+ 2.5 Attempted calculate but not 1-complement of 1-complement sum or not 16-bit numbers
+ 2 Give semi-valid answer not to the question asked
+ 0 No attempt / Incorrect

5.3 5.3 Demultiplex 6 / 6

- **0 Correct**
- 0 Not mentioned that OS uses UDP-specific lookup table to find app socket
- 3 Incorrect mentioning of sourceIP/sourcePort as part of the lookup
- 1 Didn't mention destination IP for demultiplexing
- 3 Didn't mention use of destination ip&port to lookup in kernel's UDP socket-app table
- 4 Only showed port number
- 6 Incorrect / missing

5.4 5.4 UDP facts 4 / 4

+ **1 UDP payload (2^16)**
+ **1 Port numbers (0, 2^16-1), ok if start 1024**
+ **1 Number of distinct apps (2*2^16)**
+ **1 Number of apps to prevent (2^16)**
+ 0 Incorrect

QUESTION 6

6 20 pts

6.1 6.1 Secret message 3 / 4

- 0 Correct
- **1 Mentioned public key encryption, but didn't discuss how to get public key for a person you never met before**
- 0.5 Mentioned how to get public key, but didn't define why it should be trusted
- 4 Incorrect / missing
- 3 Mention PGP, but in incorrect context

6.2 6.2 Info from email 3 / 4

+ 4 Correct
+ **2 Mentioned PGP signing and telling (out-of-band) your key (key fingerprint) to the professor / out-of-band acknowledging sending email**
+ 2 At least two objective info items listed
+ **1 At least one objective info listed**
+ 0 Incorrect / missing

6.3 6.3 Invalid HTTPS 4 / 4

- **0 Correct (At least 2 reasons listed)**
- 2 Only one correct reason listed
- 4 Invalid / missing answer

6.4 6.4 Multiple DNS records for youtube.com/A 3 / 4

- 0 Correct (at least 3 correct reasons listed)
- **1 Only two correct reasons listed**
- 2 Only one correct reason listed
- 0 Incorrect/missing answer

6.5 6.5 HTTP/2 vs QUIC 4 / 4

- **0 Correct**
- 1.5 Only one reason listed
- 4 Incorrect/missing answer

CS118
Spring 2017 Midterm Exam

1 hour 50 minutes
Close book and closed notes,
except a SINGLE piece of paper as a cheat sheet.

NO use of any device except calculators.

- This exam has 7 pages, including this cover page. Do all your work on these exam sheets. NO EXTRA PIECES OF PAPER WILL BE ALLOWED.
- Cross out all the scratch work that you do not want to be counted as part of your answer before you submit the exam.
- Be *specific, clear,* ***concise*** in your answers, and *explain* your answers.
- When the answer to a problem is not immediately clear, do not simply dump everything, relevant or irrelevant, on the paper. Irrelevant answers may lead to point-deduction as they show the lack of understanding of the problem.

Your name: Karen Li

Student ID: 204564235

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

format of messages sent and received,

order of messages sent and received, and actions taken upon receipt or transmission of a message.

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 by caching resolver and replicated over many authoritative servers.

HTTP protocol can scale because WEB data can be

cached by HTTP proxy and sent quickly by using pipelining for HTTP/1.1 and HTTP/2.0

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

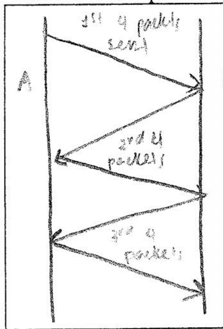
multiplexing & demultiplexing.

In addition to this function, TCP also provides

reliable, in-order byte stream delivery, 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).



Transmission Delay for 4 Packets = (10 000 bits) · 4 / (10^6 bit/s) = 0.04 s

Time to send 4 packets = (0.04 s) + (0.001 s) = 0.041 s

Time to get 4 ACKs = 0.001 s (only prop. delay since trans. time is 0)

Time for 12th packet to arrive at B = 3(0.041 s) + 2(0.001 s) = $\boxed{0.125\,s}$

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

Number of 4-packet groups that must be sent: (500 000) / ((10 000 bit/packet)(4 packets)) = 12.5 ≈ 13

We need to make 13 "sends" and receive 12 ACKs (need ACKs for all the sends except the very last one). Like in part a), the time to make a "send" is 0.041 s and the time to get ACKs is 0.001 s.

Total time = 13 (0.041 s) + 12 (0.001 s) = $\boxed{0.545\,s}$

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?

time to send 4 packets = (0.04 s) + (0.1 s) = 0.14 s

Time to get 4 ACKs = 0.1 s

Total time = 13 (0.14 s) + 12 (0.1 s) = $\boxed{3.07\,s}$

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)?

transmission Delay for N packets = (10000 bits) · N / (10^6 bits/s) = 0.01 N s

Time to send N packets = (0.01N + 0.1) s

Time to get N ACKs = 0.1 s

Number of "sends" needed = (500 000) / ((10000 bits/packet)(N packets)) = 50/N sends

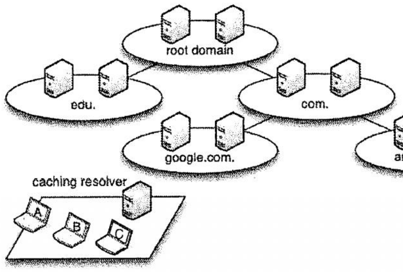
Number of ACKs needed = (50/N − 1) ACKs

Total time = (50/N)(0.01 N + 0.1) + (50/N − 1)(0.1) = 0.5 + 5/N + 5/N − 0.1

= (0.4 + 10/N) s

As N → +∞, total time approaches its minimal value, 0.4 s.

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?

To get "A record for amazon.com", stub resolver must contact caching resolver, and caching resolver must contact root domain, .com domain, and amazon.com domain: (20 ms) + 3(150 ms) = 470ms

To get "A record for www.amazon.com", the amazon.com domain is already in the cache so caching resolver only needs to query once: (20 ms) + (150 ms) = 170 ms

Total time = (470 ms) + (170 ms) = $\boxed{640 \text{ ms}}$

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?

The .com domain is still in the cache, so caching resolver queries it to get the domain for google.com, then queries that to get MX record.

Total time = (20 ms) + 2(150 ms) = $\boxed{320 \text{ ms}}$

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)?

For mail.google.com, you have google.com domain in the cache, so the caching resolver just needs to query once: (20 ms) + (150 ms) = $\boxed{170 \text{ ms}}$

For hangout.google.com, similarly, you have google.com domain in the cache so the caching resolver just needs to make one query = (20 ms) + (150 ms) = $\boxed{170 \text{ ms}}$

The time to receive each of the answers was 170 ms

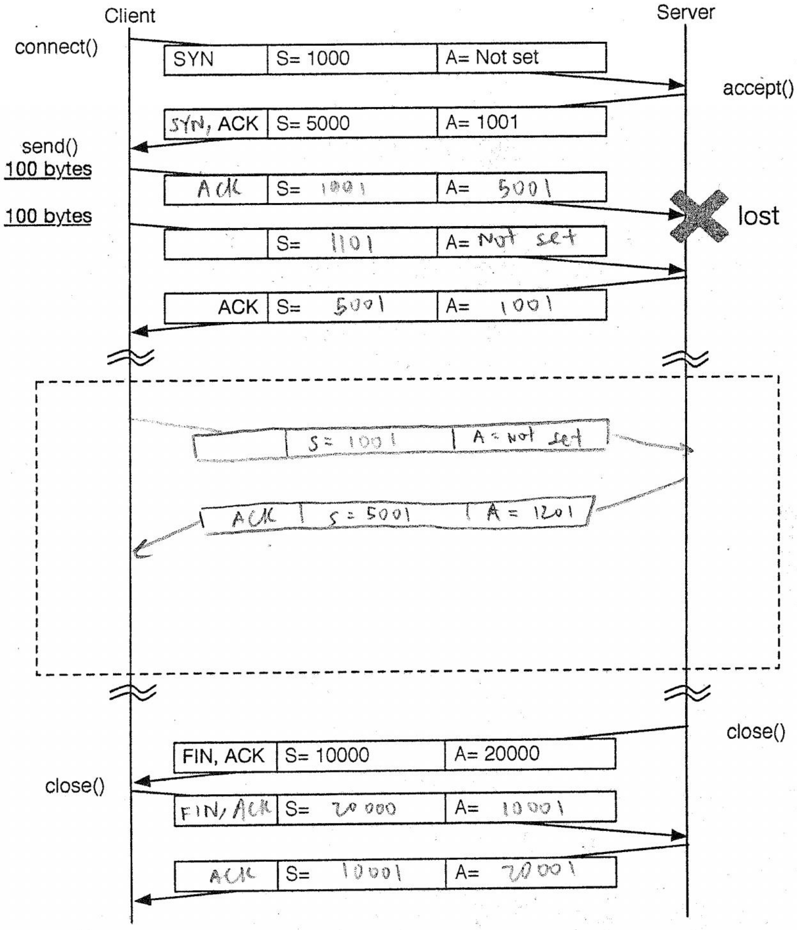
3.4 (5 points) List DNS records that the caching resolver has at T=90 minutes

The four DNS records from 3.2 as well as the "AAAA (IPv6) record for mail.google.com" and the "AAAA (IPv6) record for hangout.google.com".

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.

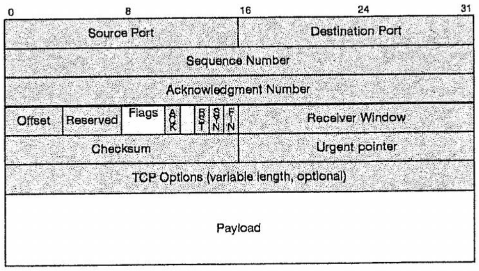


loss has been detected, so send the oldest unACK'd segment

ACK # = 1201 since we buffered the out-of-order input and are using cumulative ACKs

S= 5001 still because we did not send any data

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.



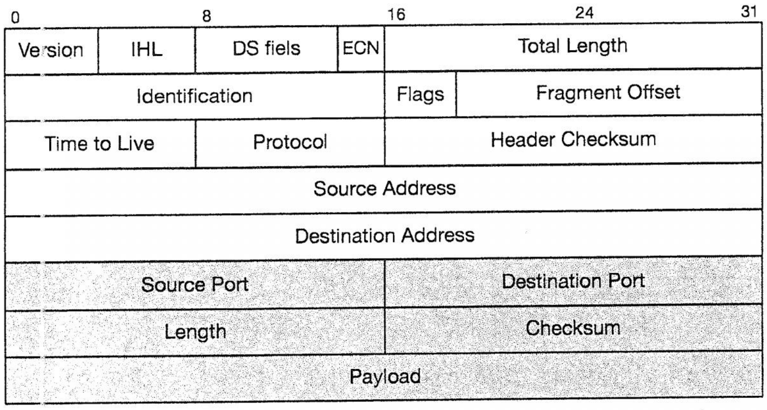
4.2 (5 points) One of the packets got lost. In the dotted box above, add the missing exchanges between the client and the server just after the loss has been detected. In the exchange, include flags, sequence number, and acknowledgement number (if applicable).

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)?

throughput = (# bytes sent) / (time needed)

time to transmit 1 packet = $(1000 \times 8 \text{ bits}) / (1000 \times 10^{12} \text{ bits/s}) = 8 \times 10^{-12}$ s

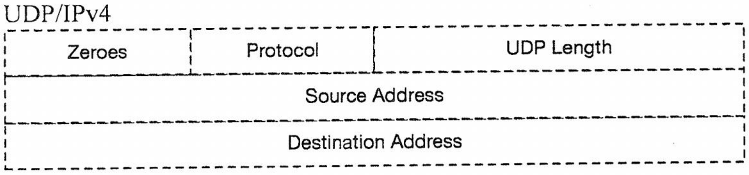
Problem 5 (20 points): The following diagram shows the UDP packet header and HEX value of one of the captured UDP packets.



```
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
```

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?

UDP checksum: one's complement of sum of UDP header/data + pseudoheader (except checksum)
(0xc26e + 0x03e8 + 0x000e + 0x4865 + 0x6c6c + 0x6f0a) + (0x0011 + 0x000e + 0x7f00 + 0x0001 + 0x7f00 + 0x0001) = X
take the top 16 bits of X and add with the bottom 16 bits of X.
Take the one's complement (flip each bit of result) to get correct checksum

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

IPv4: one's complement of sum of all 16 bit words in IPv4 header (except checksum)
(0x4500 + 0x0022 + 0x23c5 + 0x0000 + 0x4011 + 0x7f00 + 0x0001 + 0x7f00 + 0x0001) = Y
Add top 16 bits of Y with bottom 16 bits of Y.
Take one's complement (flip each bit of result) to get correct checksum.

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.

In UDP, sockets are identified by a two-tuple = (destination IP, destination port #).
Thus, the OS will send this packet to the application process with socket listening on port number 0x03e8 = $8 + 14 \cdot 16^1 + 3 \cdot 16^2 = 1000$ and with host IP equal to 0x7f000001 = 127.0.0.1 (local machine).
The application should be listening on port 1000.

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

The maximum size of a UDP payload is (2^{16} bytes) – (8 bytes header) = 65528 bytes

The range for UDP port numbers is 2^{16} = 65536. Thus, range is [0, 65535].

For a computer with two IP addresses (e.g., one for wireless and one for wired), there could be (# of ports) x (# IPs) = (65536)(2) = 131072 maximum number of distinct UDP server applications.

To prevent anybody else to start a UDP server application, one need to start at least 65528 number of applications, each creating one socket, binding, and listening on a single port.
(can use 0.0.0.0 as wildcard destination IP).

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, you can send a secret message over email by encrypting the email with that person's public key. That way, only that person can read the email since they are the only person with the private key able to decrypt the email.

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.

1. The time that the email was sent.
2. The time the email was received.

} cannot be faked

To ensure that the email is from me, I should sign the email using my private key so that when the professor opens the email, he will see that my public key can decrypt the signature, indicating that the email is from me.

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. certificate could have expired
2. certificate could be revoked by CA.
3. server could be untrusted by the CA so certificate is invalid.
4. certificate info could have been corrupted on server side.

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.

1. Different locations have different A records in caching resolvers.
2. If not in cache, different locations can query different root / TLD servers, will get different results.
3. There are multiple A records for youtube.com on an authortative server.
4. Different location CDNs might send users different A records to balance load or reduce network distance.

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.

1. QUIC runs over UDP and not TCP so application has more flexibility for managing congestion control and flow control.
2. No head-of-line blocking in the transport layer, while HTTP/2 still has some head-of-line blocking in transport layer.