CS118 Midterm

TOTAL POINTS

87.5 / 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 ... 1 / 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 2 / 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 0 / 6

+ 6 Correct

+ 3 Correct transmission delay

✓ - 1 Incorrect number of RTTs

+ 0 Incorrect

2.2 2.2 Total delay 0.75 / 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 3 / 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 4 / 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 2 / 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 3 / 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 2.25 / 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 2.5 / 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 2.5 / 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 3 / 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 2 / 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 4 / 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: _____

Student ID: _____

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

how to communicate, how to deliver data, and how to determine resource availability.

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

be cached and replicated.

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

multiplexing, demultiplexing.

In addition to this function, TCP also provides

congestion control, flow control and connection establishment.

500000/10000 = 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).

$\frac{12}{4} \cdot 1\ ms = 3$ milliseconds

$\frac{500000 \text{ bit file}}{10000 \text{ bit packets}} = 50$ packets

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

$\frac{50}{4} \cdot 1\ ms = \frac{25}{2} \cdot 12.5$ milliseconds

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?

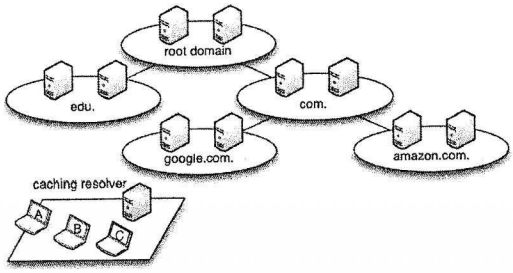
$\frac{50}{4} \cdot 100 = 1250$ milliseconds

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)?

Link capacity = 1 Mbps → 10^6 bits/s = $10^6/10000$ = 100 packets/second

$\frac{50}{50} \cdot 1 = 1$ millisecond, window size of 50

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?

cache(20) + root(150) + com(150) + amazon.com(150) + cache(20)

= 490 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?

query .com for google.com (150) + query google.com (150) + cache (20)

= 320 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)?

a) cache(20) + root(150) + com(150) + google.com(150) = 470 ms

b) google.com(150) + cache(20) = 170 ms

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

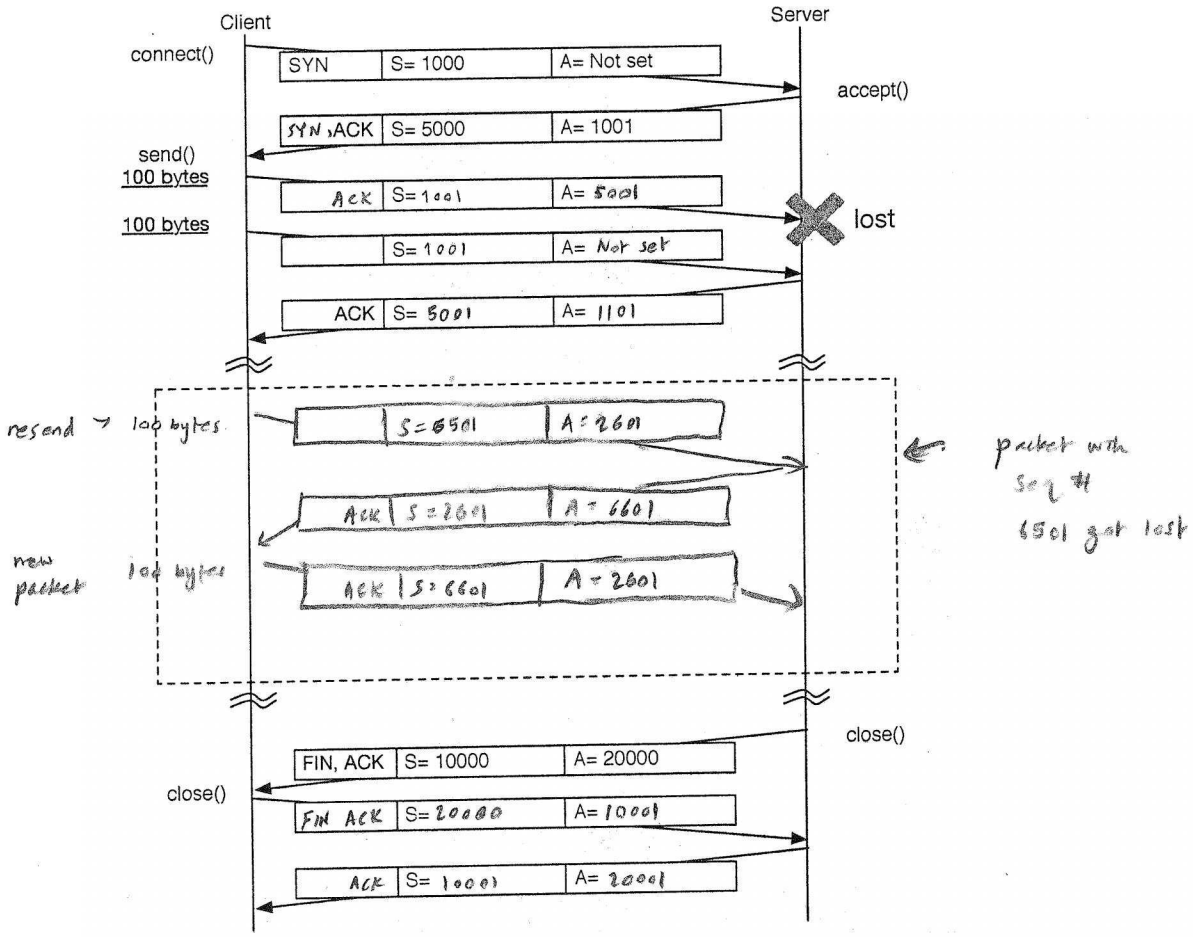
NS for root, NS for .com, NS for google.com

AAAA records for mail.google.com, hangout.google.com

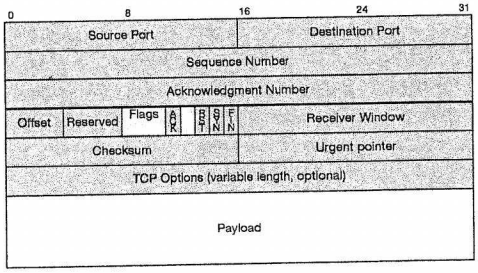
3.5 (3 points) At T=100 minutes, all the authoritative servers for **.com** go offline. Circle the domain names that can be resolved by Host-A?

(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.



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)?

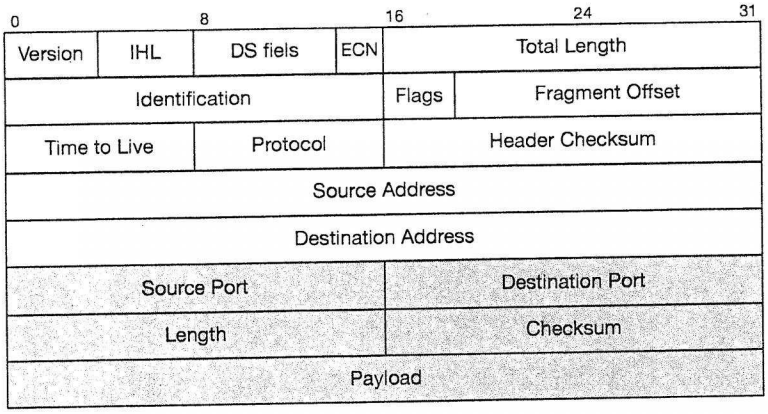
throughput = 1000 bytes → 8000 bits / packet

$\frac{8000 \text{ bits}}{500 \text{ ms}} = \frac{8000 \text{ bits}}{0.5 \text{ s}} = 16000 \text{ bits/s}$

→ $\frac{1000 \times 10^{12} \text{ bits/s}}{16000 \text{ bits/s}}$

$= 6.25 \times 10^{10}$ packets

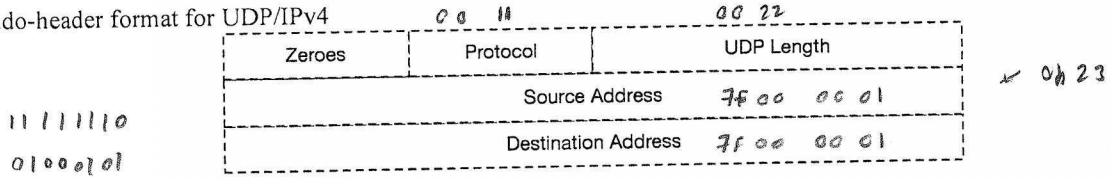
Problem 5 (20 points): The following diagram shows the UDP packet header and HEX value of one of the captured UDP packets.




```
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 = UDP header + segment + pseudo header

= No, 0hdc

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

```
11111110
00100100
00100011  → 0h23
```

complement

11011100 → 0hdc → 0h30

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.

On UDP, sockets can be fully identified with destination IP and dest port. In this case the destination IP is given by hex 7f00 0001 and port is 03e8 which is 1000.

127.0.0.1

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

The maximum size of a UDP payload is 4 bytes

The range for UDP port numbers is 65536

65536·2

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

To prevent anybody else to start a UDP server application, one need to start at least 65536 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?

You can encrypt your email with either symmetric key or the recipient's public key. The recipient can decrypt it with symmetric key or his/her private key. For authentication you can send a digital certificate with your 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) If you signed it with your private key, he can use your public key to verify that the email is from you. (to create a digital signature)
2) But anyone can share their public key and pretend to be you. Solution: use certificate authority such as PGP to prove that the public key is yours.

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 certificate
2) Revoked certificate
3) Wrong host (applicant may not have right to access domain name)
4) Untrusted root (root may not be trusted)

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.

"Abuse" of DNS because of 1) replicated authoritative servers 2) IPs are changed for security purposes, 3) CDNs may manage servers in multiple regions around the world 4) caching resolvers may die (e.g. Web portal)

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.

↓ doesn't have to establish connection
QUIC runs over UDP and has app-controlled congestion control. It has multiplexing and flow control similar to HTTP/2 and also has security equivalent to TLS. Also has forward error correction, connection migration and server push.