Instructions

Please answer each question with a written response (no code required). Feel free to use whatever software you want for your write-up. However, you should convert it to a `pdf` file for your submission upload.

Exam rules:

- You have 24 hours for the exam (8:00AM ET Friday to 4:00PM ET Saturday).
- You **can** use the textbook, your notes, previous lectures, and the internet.
- You **cannot** discuss the material with other students during the exam. Additionally, you cannot ask questions in an online forum (e.g., stack overflow). If cheating is detected, you will receive a 0 on the exam and the case will be forward to Office of the Dean of the College.
- If you have any questions at all, please email me! I will post clarifying announcements that may arise from this questions on Friday.

This exam is worth 20% of the total grade. Quickly look through the exam being mindful of the point breakdown and plan your time use wisely. The point total is 100.

1. Bandwidth and Latency (14 points)
2. Custom top-level domain in DNS (14 points)
3. HTTP Multiplexing (14 points)
4. Distributed Hash Table (14 points)
5. Custom Transport Protocol (16 points)
6. TCP Congestion Control (14 points)
7. Packet Forwarding (14 points)

1 Bandwidth and Latency (14 points)

Consider the following two means of transmitting information between two hosts that are 4,000 kilometers apart (e.g., from Lancaster, PA to San Francisco, CA).

- **Copper wire**: with a propagation delay of 4,000,000 m / (200,000,000 m/s) = 0.02 seconds and a bitrate of 10 mbps (10,000,000 bits per second).
- **Carrier pigeon**: with a propagation delay of 4,000,000 m / (16 m/s) = 250,000 seconds and each pigeon can carry a 4TB flash drive. As soon as the pigeon arrives, the 4TB of data can be considered delivered.

Assuming no loss of data (or pigeon), explain which mean of transmission will deliver **the entire message faster** in the following two scenarios:

(a) A message containing a 32KB text file.

(b) A message containing 3TB of YouTube videos.

Your answer must include calculations for both transmission mediums in each scenario for credit.

2 Custom top-level domain in DNS (14 points)

You decide you want to make your own top-level domain (TLD), `.coffee`, as you envision some innovative new business idea with coffee domains (e.g., `brad.coffee`). Answer the following questions about how your TLD would be set up and operate:

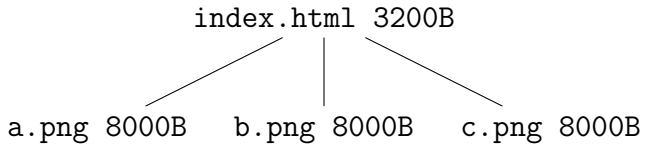
(a) Who must you convince to enable users to make DNS requests for your top-level domain?

(b) How a user would get an IP address for a HTTP web-server associated with `brad.coffee`?

(c) Unlike better funded TLDs, you can only purchase and maintain 3 `.coffee` DNS servers. Assume these are geographically distributed around the earth (one in US, one in France, one in Singapore). What are the potential implications to new users requesting `brad.coffee` if any of the DNS servers go down?

3 HTTP Multiplexing (14 points)

HTTP multiplexing allows for a single TCP connection to be shared across multiple HTTP requests. At a high-level it works like this (refer to requested files below): a client can request multiple files, e.g., `a.png` and `b.png`, one immediately after another using the same TCP connection without waiting for the server to deliver `a.png` first. Note that no children can be requested (e.g., `a.png`, `b.png`, `c.png`) until the parent (e.g., `index.html`) is received by the client.



To understand the potential benefit of multiplexing, compute how long it take for a client to fetch all resources in the figure below under the following scenarios:

(a) A single TCP connection where each file must be fully delivered before requesting a new one.

(b) A single TCP connection with multiplexing (files can be requested one after another).

Make the following assumptions for both scenarios. The transmission rate is 8 mbps (megabits/second) from both client-to-server and server-to-client. **The client-to-server and server-to-client latency is 100ms.** Assume no packet loss or bit error. Each HTML request/response header is 26 bytes.

Finally, use these calculations to explain why latency plays such a large role in the benefits of multiplexing.

4 Distributed Hash Table (14 points)

In class, we talked about a peer-to-peer (P2P) network for distributing files among peers without needing a centralized server. Another common P2P application is a distributed hash table (DHT) which provides a lookup service for key-value pairs that are stored across many peers. Typically, each peer has only a subset of the key-value pairs. When a peer in the P2P network wants to look up a key, it asks one or more peers with the key-value pair for the value. Based on this framework, please answer the following questions (a high-level explanation is fine):

(a) What problem(s) could emerge if a peer could update its own local hash table without notifying other peers in the network?

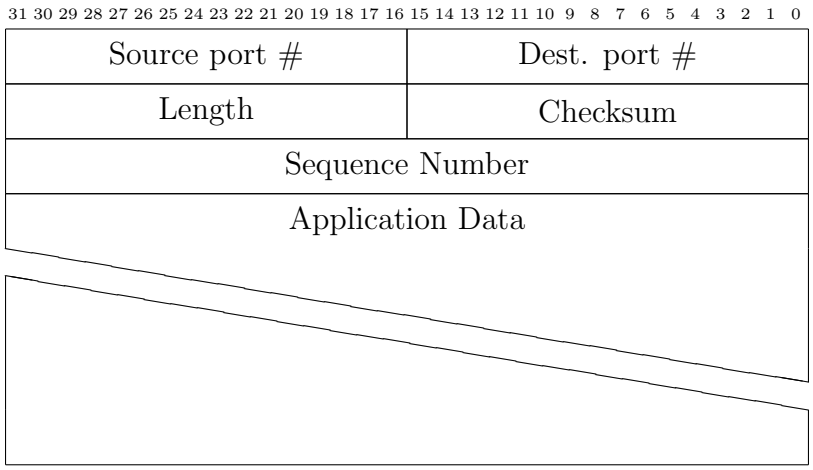
(b) What are the potential security implications of trusting random peers?

(c) Does querying multiple random peers for the same key-value pair reduce the security concern?

5 Custom Transport Protocol (16 points)

You have the brilliant idea of implementing a new transport protocol, with a carefully market-tested name, called Somewhat Reliable User Datagram Protocol (SRUDP). The goal of the protocol is to get most of the reliable transmission guarantee of TCP without the need for ACKs. The protocol will still use sequence numbers to provide in-order delivery. Instead of sending each packet a single time, SRUDP sends the same packet 5 times (back to back) to increase the chances that one of the packets makes it through. The SRUDP segment structure is given as follows:

SRUDP segment structure



Unfortunately, there are many issues with this design. Answer the following:

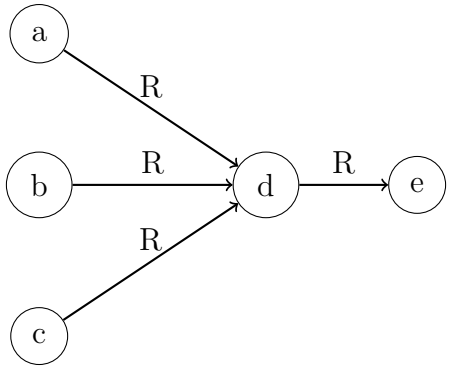
(a) What problem will SRUDP face in regards to network congestion? Hint: related to sending each packet back-to-back.

(b) SRUDP still wants to provide in-order delivery of packets, for all packets that are delivered successfully (sometimes all 5 packets will be dropped). Describe a solution for how SRUDP could decide to hand off data to the application given that some packets will still drop?

(c) What other problems are raised by your solution in (b) for in-order delivery?

6 TCP Congestion Control (14 points)

TCP congestion control ensures fair utilization of a shared link. In the graph below, three nodes (a, b, and c) want to transmit to e through d at a rate of R bits per second. However, since the link connecting d and e can only transmit at a rate of R, nodes a, b, and c must settle for a lower transmission rate.



Based on the principle of fair utilization, answer the following questions about this graph:

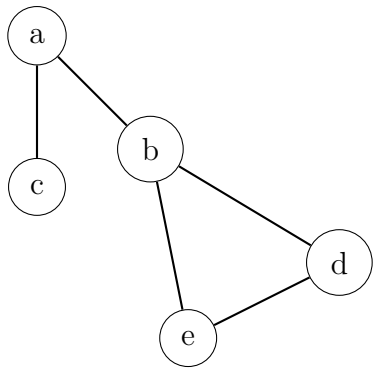
(a) Under TCP, what is the achieved throughput of a, b, and c, respectively.

(b) Assume b stopped sending packets, what is the achieved throughput of a and c.

(c) What (if anything) prevents c from cheating and opening multiple TCP connections in order to achieve a higher (but less equitable) throughput?

7 Packet Forwarding (14 points)

Packet forwarding is the local decision a router (node) makes on where to send a packet based on its destination. Below is a graph of 5 nodes with an associated forwarding table for each node. For instance, the node a table says if a packet has a destination (dest) of d then it should be forwarded (next hop) to b. For this problem, assume that the hop count for any packet is initialized to 5, is decremented each hop, and is dropped when it reaches 0. Also assume that these forwarding tables are static and will not be updated.



Node a Table

Dest	Next Hop
b	b
c	c
d	b
e	b

Node b Table

Dest	Next Hop
a	a
c	d
d	d
e	e

Node c Table

Dest	Next Hop
a	a
b	a
d	a
e	a

Node d Table

Dest	Next Hop
a	b
b	b
c	e
e	e

Node e Table

Dest	Next Hop
a	b
b	b
c	b
d	d

For the above graph and forwarding tables, explain how each packet below would propagate through the network:

(a) A packet originating at c with destination e.

(b) A packet originating at e with a destination c.