

# Problem Set 1

## 15-440/15-640 Distributed Systems Spring 2025

**Assigned:** Thursday, January 23, 2025

**Due:** Thursday, January 30, 2025 at 11.59 pm

### Submission procedure:

- Create a **.pdf** of your answers and upload to [Gradescope](#). All enrolled students should have Gradescope accounts. Ask **well in advance of the deadline** to be added to Gradescope if you don't have access; it is unacceptable to ask to be added immediately before the deadline.
- Here are some tips to make your submission easy to read and to grade. Remember, the easier you make this, the less likely we are to make grading errors. Following these guidelines will help us to focus on the technical content of your answers rather than trying to understand what you have written.
  - Don't hand write your answers. Use Latex or Google Docs or some similar input mechanism. If you use Latex, a template can be found on the course web page.
  - Put the answer to each question on a separate page.
  - Carefully tag your pdf pages to each question on [gradescope](#). You can use the SHIFT key to select multiple pages and associate them with a single question.
- Assume SI notation
  - 1 KB =  $10^3$  bytes, 1 MB =  $10^6$  bytes, 1 GB =  $10^9$  bytes
  - 1 Kbps =  $10^3$  bits per second (bps), 1 Mbps =  $10^6$  bps, 1 Gbps =  $10^9$  bps
  - but a byte is still 8 bits (not 10 bits) :-)
- **Remember that you have a limit of 2 grace days totaled over all four problem sets.** You can use at most one of those grace days per problem set. Although Gradescope does not track grace days, we will. Exceeding your grace days will result in a zero grade for that problem set.

## Question 1 (30 points)

Ruiqi, a creative mind at CMU, has devised a new game that is played on a 100 x 100 board. Each square of the board can take a value between 0 and 100. The board starts out with all zeros. The players take turns: each player makes a local move, and then sends the updated board to the opponent. The detailed rules of a move are not important for this question. All that matters is that a single move may change multiple, possibly non-adjacent, squares of the board. A monotonically increasing counter, that starts at 1, keeps track of the move number as the game proceeds. Ruiqi has invited her friends Michael and Jimmy to test the game.



Ruiqi has contributed the pseudo-code shown below.

```
typedef struct gamestate {
    int8_t* squareval; // The 100 * 100 game board
    int32_t round; // The round number
} game_t;

// This code performs marshalling of game state; some details are omitted.
char *serialize(game_t *currentstate) {
    /*
    Skipped code that defines variables ...
    */
    size_t total_size = (1);
    char *data_packet = malloc(total_size);
    memcpy(data_packet, total_size, (2));
    // Include the board
    int offset = (3);
    for (int i = 0; i < 100; i++){
        // copy row i
        size_t row_size = (4);
        memcpy(data_packet + offset + (i * row_size), (5), row_size);
    }
    /*
    Skipped code that includes round ...
    */
    return data_packet; // packet ready for transmission
}
```

- A. Fill in the missing code fragments for `serialize` that are highlighted above using notation such as “`(1)`”. Assume the application is running on a 64-bit system. Include a short explanation for each piece of missing code.
- B. Suppose that marshaling and unmarshaling stubs take **5 ms** each on the client and server. Further suppose that the network has a one-way latency of **10 ms**, and a bandwidth of **240 kbps**. Calculate the total delay from the moment Michael completes his move, to the moment Jimmy can start thinking about his move. Show your work.
- C. Realizing that transmitting the entire board state can be costly in time, Ruiqi asks you to help her more efficiently transmit moves. Suppose each move can modify at most 10 squares of the board. What is your advice to Ruiqi? Give an informal explanation first, then show the data structure that you would use to transmit a move.
- D. Suppose the average number of board squares that change per move is 5. However, the marshalling/unmarshalling times of the stubs do not change. Compute the average size of data transmitted per move, and the average time delay for transmitting a move using
  - a. the original encoding of game state, and
  - b. your more efficient encoding of game state.
- E. Suppose the rules of the game are changed dramatically. As a result, the average number of board squares that change per move is 5000 rather than just 5. How does your answer to (D) change?
- F. Based on your calculations for (D) and (E), what insight can you offer about efficient maintenance of shared state between a client and a server?
- G. After many months of happy game playing with the original code (i.e, as shown in the pseudo-code above for (A)), Jimmy upgrades his computer. Unlike his old machine, which was based on an Intel chip, the new one is based on a non-Intel chip. Alas, his games with Michael after the upgrade are disastrous. The games are totally messed up, and sometimes involve the game software crashing for Michael or Jimmy. They are dazed and confused. Can you help them? How could the hardware upgrade have resulted in such a problem? How can the problem be fixed without any further hardware changes?

## Question 2 (20 points)

Helen is a boba recipe creator. After a prolific session of creating many new recipes, she needs to send them all to a central boba server for safekeeping. She is considering two different protocols:

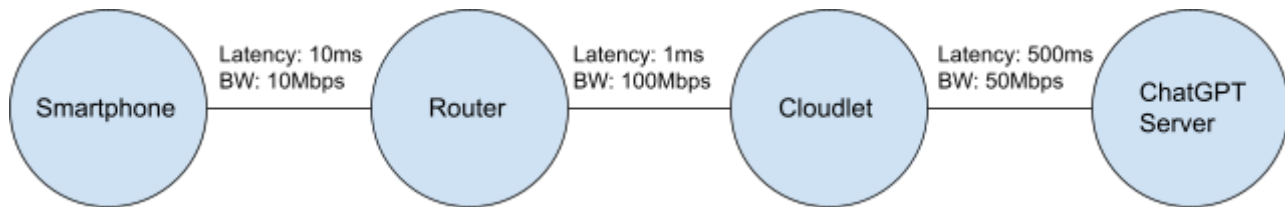
- **Stop-and-go protocol:** Helen sends a single recipe and then waits to receive an ACK from the server. She waits for at most  $T$  seconds. If no ACK arrives within time  $T$ , Helen retransmits that recipe. She repeats this indefinitely, until the ACK arrives. She then proceeds to send the next recipe, and so on until she receives the ACK for the last recipe.

- **Blast protocol:** Helen sends all the recipes at once and waits to receive a single ACK from the server. If no ACK arrives within time  $T$ , Helen will retransmit all the recipes again. She repeats this indefinitely, until the ACK arrives.

Assume for questions (A) through (C) below that the network is perfectly reliable. You may also assume no processing delay at the server (i.e., an ACK for a received recipe is sent instantly), and that ACKs are negligible in size. Each packet is  $B$  bytes long and contains exactly one recipe. For each of the situations (A) through (C) below, calculate how long it would take for Helen to send  $M$  recipes to the boba shop using the two different protocols.

- Helen is at a crowded cafe near the boba shop with a one-way latency of 10 ms and a bandwidth of  $B$  bps.
- Helen is on Mars with a bandwidth of  $B$  bps and one-way latency of  $L$  seconds.
- For each of the above situations, which protocol is better? Explain why that is the case.
- For question (B), let's remove the assumption of perfect network reliability but leave everything else unchanged. If the probability of packet loss is high, which protocol should Helen choose? Explain your reasoning.

### Question 3 (20 points)



Consider the following end-to-end pipeline for queries from your smartphone to ChatGPT.

- The query travels on Wi-Fi from your smartphone to a router.
  - Latency (one-way, symmetric): 10 ms
  - Bandwidth (symmetric): 10 Mbps
- The router forwards your query over Ethernet to a nearby cloudlet.
  - Latency (one-way, symmetric): 1 ms
  - Bandwidth (symmetric): 100 Mbps
- The cloudlet processes the query on its own if it is able to, and responds without contacting ChatGPT. If the cloudlet cannot process the request on its own, it forwards it to the ChatGPT server. It makes this decision in negligible time. For all of the questions (A) through (D) below, you may assume that the cloudlet does indeed have to forward the request on to ChatGPT (i.e., it is unable to process the request by itself).
  - Latency (one-way, symmetric): 500 ms
  - Bandwidth (symmetric): 50 Mbps

- The ChatGPT server takes 200 ms to service any query, regardless of size.

For the questions below, you may assume that all networks are ideal: i.e., there is no loss, corruption or reordering of packets. Show all your work and don't round any values until the very end. Express your final answer in milliseconds.

- Suppose that a query is 10 KB in size and the response is 5 KB. What is the total end-to-end delay of this pipeline?
- ChatGPT also supports complex queries via file uploads. You upload such a complex query as a 30 MB file, which is sent as a back-to-back stream of 1000 equally sized packets (i.e. each packet is of size 30 KB). After receiving the entire file, the cloudlet realizes that it has to pass it on to ChatGPT. It again transmits the file as a back-to-back stream of 1000 equally sized packets to ChatGPT. After ChatGPT runs its model inference, it returns a response of size 5 KB. What is the total end-to-end delay for such a query?
- Suppose the university upgrades the LAN connecting the router to the cloudlet from 100 Mbps to 1 Gbps. Nothing else is changed. What is the total end-to-end delay for (B) now? Explain your answer.
- Suppose you use Ethernet to directly connect your smartphone to the cloudlet. So you are not using WiFi at all, and avoid use of the router. The smartphone to cloudlet connection has 10 ms latency and 1 Gbps bandwidth. What is the total end-to-end delay for (B) now? Explain your answer.

#### Question 4 (15 points)

The US Postal Service creates a new service for secure and guaranteed delivery of mail. To use this service, the customer needs to provide the contents of the letter as a human-readable document, and (separately) its destination address. An encryption machine in the receiving post office encrypts the contents of the letter, prints out the encrypted text using a printable encoding, puts that into an ultra-secure envelope to the addressee, and then seals the envelope. The original provided by the customer is shredded. That sealed envelope is transported, without opening, through the multi-hop nationwide Postal Service routes. Unlike the flimsy envelopes used by customers that are prone to splitting open and spilling their contents, these sealed envelopes are virtually indestructible even in the worst conditions.

At the destination post office, the steps are reversed: the envelope is opened, the contents are decrypted by a machine, and then printed on to a document. The human-readable printed document is now placed in an envelope to the addressee, and sealed. The mail carrier delivers this envelope to the addressee.

For each of the following scenarios, state whether the US Postal Service is violating the end-to-end principle or not. In each case, explain your reasoning.

- A. Caitlyn wants to order a new pair of shoes from FakeNike. She fills out a paper order form with the name of the shoes, her address, credit card details, etc. She then sends the filled-out form through the mail service described above. Later that day, she sees that the shoes have been further reduced in price, and decides to buy a second identical pair. She therefore places an identical order via the mail service. At the end of the day, when the receiving post office processes all its incoming mail, it notices two identical letters to the same addressee. In the interests of efficiency, it discards one of them. Think of the carbon footprint it has reduced by avoiding transport and delivery of the duplicate order! FakeNike only receives one order, and Caitlyn only receives one pair of shoes.
- B. Jayce wants to send Viktor the latest blueprints of his Super Secret Enchanted Explosive Device. He does not trust the Postal Service's new service that was described above. He therefore uses old-fashioned standard Priority Mail. To ensure that the blueprints are useless even if they fall into the wrong hands, Jayce uses a magic binding spell that only Viktor's unique magic can unbind. The spell-bound blueprints are packaged in a Priority Mail box, and then handed to the Postal Service. The postal service does not tamper with or inspect the package. The package reaches Viktor safely and intact, but badly delayed due to the extreme cold weather en route. At the destination, Viktor opens the box, unbinds the magic spell, and is able to view Jayce's blueprints.

### Question 5 (15 points)

A new social media app Outstagram, created by CMU Qatar students, uses a centralized server located in Qatar for all of its users and data. It allows users to post and share pictures with each other via RPC to the server. Upon receiving an uploaded image, the server will store it in a database. Although initially designed for users local to Qatar, it has quickly gained popularity at CMU Pittsburgh and built a large user base here as well. Unfortunately, these CMU Pittsburgh students are experiencing frequent long loading times while using Outstagram. They sometimes see an error screen saying "TCP timeout" when they try to upload pictures.

- A. Give 3 plausible and distinct reasons for the slow loading times.
- B. It turns out that even Qatar users are experiencing poor performance despite excellent nationwide LAN and WLAN connectivity within Qatar. Which of your 3 reasons in answer (A) are still plausible reasons now? Explain your answer.
- C. A self-assured and expensive consultant tells the app developers that the slow performance is due to the old-fashioned magnetic disks that are being used to store the image database. At high cost, the developers replace these magnetic disks with flash drives. Alas, no improvement is noticed by anyone, even those in Qatar. Does this fact narrow the set of plausible explanations for the slow performance? Explain why or why not. Based on all this evidence, what is the most likely root cause of the poor performance?