Problem Set 1

15-440/15-640 Distributed Systems Spring 2024

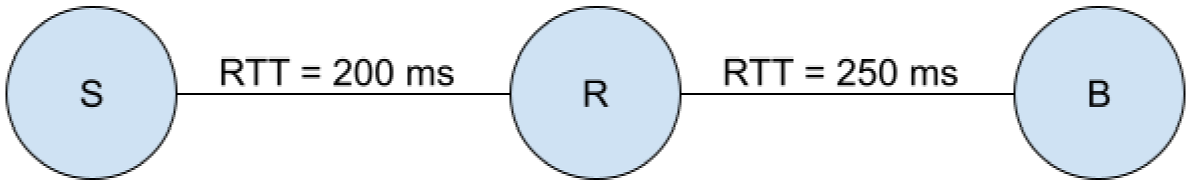
Assigned: Thursday, January 25, 2024

Due: Tuesday, February 6, 2023 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 (20 points)



Consider 3 nodes S, R and B that are connected by the network links shown above. The RTT of the S-R link is 200 ms, and the RTT of the R-B link is 250 ms. You may assume that the network is ideal: i.e., there is no loss, damage or reordering of packets.

A. Two packets that are each of size 1 Kb (kilobit) are sent back-to-back from R to S. An ACK is sent by S for each packet as soon as it is received. R receives the second ACK 0.001 ms after it receives the first ACK. What is the bandwidth of the S-R link?

B. When the experiment in (A) is repeated between R and B, the delay between the two ACKs is measured to be 1 ms. What is the bandwidth of the R-B link?

C. A file of 10 Mb is sent from R to S as a back-to-back stream of 1000 packets, each of size 10 Kb. After the last packet is received by S, there is a delay of 15 ms, and then a single ACK is sent back to R. Assuming the network bandwidth of the S-R link is 100 Mbps, what is the total file transfer time from R's viewpoint?

D. R, being spiteful, wishes to painfully slow the transmission of a 10 Mb file that is sent from S to B via R. The RTT of the S-R link is increased to 500 seconds (not milliseconds), but its bandwidth remains 100 Mbps. However, the RTT of the R-B link remains unchanged. The file is transmitted in packets of size 10 Kb, and a single ACK is sent by B to S after the last packet is received. How low should the bandwidth of the R-B link be so that the file takes one full day (86400 seconds) to be transferred?

   (NOTE: Show all your work and don't round any values until the very end. Express your final answer with three digits past the decimal point.)

Question 2 (15 points)

Princess Leia needs to obtain a crucial Death Star specification file in her fight against the Galactic Empire. The specification file is small (just a few kilobytes), and is securely stored on a Galactic Empire server. Princess Leia has penetrated and compromised this server. She initiates an RPC request over the intergalactic network to get the specification file from the server.

Suppose the intergalactic network is ideal, meaning that no packets are dropped, mutilated or reordered. You may assume that the one-way wireless network latency has a fixed value of 100 milliseconds, and that bandwidth is 1 Gbps. The RPC timeout value is set to 30 seconds.

A. Alas, in spite of the excellent network connectivity, tiny file size, and generous timeout value, Princess Leia's RPC times out. Explain two plausible reasons for this.

B. Puzzled, Princess Leia reconnects to the server, and tries the RPC again. Amazingly, the RPC succeeds this time, and she obtains the Death Star specification file. Give a plausible explanation for how this RPC could succeed, even though the identical RPC failed earlier.

C. Buoyed by her success in obtaining the Death Star specification file, Princess Leia decides to obtain the full Autodesk drawing file, which is 10 GB in size. Alas, Princess Leia's RPC times out. Repeated attempts to perform the RPC after re-establishing connectivity to the server all result in timeouts. Give a plausible explanation for why these RPCs all time out.

Question 3 (25 points)

A. You want to upload a video to your favorite video streaming platform, MeTube. The video is very long and you have cheap, low-bandwidth Internet service to your apartment. The video will take many hours to upload. Under the covers, the MeTube browser code relies on TCP to transmit user video data to its servers. After hitting 'Upload', you go to bed without waiting for the multi-hour upload to complete. Alas, there is a severe thunderstorm overnight while the upload is in progress. Internet connectivity to your home is broken for an hour, but resumes after that. In the morning, you see the message "Upload successful" on the screen. Which action semantics is being exhibited by the MeTube browser code: exactly-once, at-most-once, or at-least-once? Briefly (in a sentence or two) explain your answer. Using pseudo-code, sketch how the MeTube browser code could be implemented on top of TCP. Clearly state any assumptions you make.

B. Cameron was not paying attention when the TAs told the class "AUTOLAB IS NOT A DEBUGGING TOOL". Cameron also heard rumors that Autolab uses randomized testing to test submissions, and averages your scores across submissions. So Cameron, overly-confident in his P1 code, submits to Autolab 15 times in a row, hoping to score a 60/60. He ends up scoring 3/60 each time. Though sad, he finds solace in the fact that he just witnessed idempotency first-hand! He thinks that since both the input and the output have stayed the same across each submission, this action (submission to Autolab) is idempotent. Is Cameron right or wrong? Explain your answer.

C. A new course registration platform for CMU students is being designed. As the lead designer, you have taken 15-440 and know all about action semantics and RPC. In your browser-based client implementation, you would like the code wrapped around the RPC to be as short and simple as possible. To achieve this goal, identify the <u>*weakest*</u> RPC semantics that would be adequate for each of the actions below. Explain your answers briefly and state and justify any assumptions. We are only looking for a few sentences each.

(*Hint: At-Least-Once < At-Most-Once < Exactly-Once, where "<" indicates "weaker than")*

a. **Course Search:** Students can search for course information via the course number. The response gives details such as units, prerequisites, teaching staff, etc.

b. Course Planning: Relative to their planned schedule for the semester, a student can add or remove a course. This is not a formal course registration or course drop in the registrar’s database. It is merely for helping students to plan their schedules. However, the university monitors the number of students who seem to be interested in the course so that the right number of instructors and TAs can be allocated.

c. **Course Schedule Registration:** When the student’s schedule for the semester is ready, he or she presents it for registration. This enters a FIFO queue, awaiting attention by staff. After processing by course staff (which may take many hours or even a few days), a single email is sent to the student. This email indicates which courses were successfully added, and which ones were not. For the courses that were successfully added, the students’ bank account is charged for the relevant number of units. For courses that could not be added, an explanation is included. Students must contact CMU HUB to resolve these errors.

Question 4 (20 points)

Erin is excited to be playing in an upcoming Battleship tournament. To improve her skills, her friend David has created a simple mobile game that involves a 20x20 game board and five non-overlapping ships of size 1x3 or size 3x1. David will hide the five ships, and Erin’s objective is to land as many hits as possible within 20 turns by attacking some square on every turn. David will reply to each of her attacks by acknowledging either a hit or a miss.

The code running on Erin’s computer looks as such:

```
int32_t ErinBoard [20][20];
// this board is cleared before each guess;
// it contains Erin's attack (before RPC) and result (after RPC)
// encoding of squares is as follows:
// 0: blank, 1: current attack, 2: attack was a hit 3: attack missed

typedef struct Move {
  int32_t x,y;
} move_t;

  omitted code here to generate an attack

void attack(move_t *curr_move) {
  ErinBoard[curr_move->x][curr_move->y] = 1;
      code omitted here that sends  the  ErinBoard via an RPC to David;
      the reply to the RPC is also an ErinBoard, with “1” replaced by “2” or “3”
}
```

The code running on David’s computer looks like this:

```
int32_t DavidBoard [20][20];
// contains locations of hidden ships

code omitted here for game initialization, etc.

void processAttack () {
// invoked by server stub for attack() RPC
// receive ErinBoard from Erin
// search for the 1, find its coordinates (row, col),
// then check against DavidBoard to see if hit or miss
if (hit) {
    ErinBoard[row][col] = 2;
}
else {
    ErinBoard[row][col] = 3;
}

code omitted here that replies to RPC with modified ErinBoard
}
```

A. Calculate how many bytes are transmitted in the request and reply of each `attack()` RPC. State and explain any assumptions you make.

B. David’s friend John suggests that he should start encoding the values of the board as `char`, not `int`. Should David implement John’s suggestion? If so, how many bytes would be sent now in each RPC call? If not, explain why.

C. Suppose David wishes to optimize `attack()` RPCs even further. Assume that the header formats of RPC and networking layers are rigidly fixed, and cannot be changed. The only part of an RPC request or reply that can be optimized is the body (which contains the serialized request or reply parameters). What is the theoretically fewest number of bits that are needed in the bodies of the RPC requests and replies?

D. David contemplates a new system of playing the game. Rather than having `attack()` RPCs, he considers having a single RPC at the beginning of the game to send the entire DavidBoard solution to Erin. Then, Erin’s computer would locally process/verify attacks. Describe the advantages and disadvantages of this system relative to the finalized system from Part C, keeping in mind network latency and amount of data sent. State and explain any assumptions you make.

Question 5 (20 points)

Sam deploys his own private Large Language Model (LLM), GTP, as a cloud service using a data center located in Asia. Since the GTP project is still in the testing stage, Sam plans to only serve a few CMU students. The cloud service works almost like ChatGPT: using RPC, a user sends a prompt from his/her computer to the server via the Internet; the server processes the prompt and generates a response consisting of words, or "tokens"; server waits until all tokens are generated, before sending them as the RPC response back to the user.

A. Soon after Sam starts running his service, he receives complaints from users about unbearably long delays. Sam believes this is due to insufficient bandwidth at the cloud server. He upgrades bandwidth from the cloud server to the Internet from 1 Gbps to 10 Gbps. Alas, users complain that they see no improvement in performance. Give a plausible reason why increasing this bandwidth does not help with the end-to-end delay in this scenario.

B. Puzzled by the complaints, Sam decides to measure and profile the time spent in each part of the end-to-end pipeline so that he can identify the bottleneck. He observes that the processing time of his GTP model consists of two parts: a fixed processing time of 10 milliseconds for interpreting the user prompt, and a much longer time for generating the output tokens. The GTP model spends 50 milliseconds to generate one output token. 90% of the responses generated by the GTP model have a length of 60 to 200 output tokens. For a test prompt to the cloud server, Sam gets the following measurements:
   - Sam's prompt size: 1 KB
   - Sam's bandwidth to the Internet: 200 Mbps
   - Number of GTP output tokens in the response: 100
   - GTP response size: 2 KB
   - Cloud server bandwidth to the Internet: 10 Gbps
   - Total (two-way) end-to-end delay: 5470 milliseconds

   What is the total processing time on the cloud server? Show your calculations.

C. Assume infinite bandwidth, zero queueing and processing delay inside the Internet. Using the measurements from (B), calculate the transmission delays of Sam's prompt and the GTP response. Then calculate the network round-trip time (RTT). Show your calculations.

D. Based on your answers to (B) and (C), identify the bottleneck in the end-to-end pipeline. Select all of the following choices, with brief explanation, that could plausibly alleviate the bottleneck:
   a. Upgrade Sam's Internet bandwidth to 800 Mbps
   b. Upgrade the cloud server's Internet bandwidth to 25 Gbps
   c. Rent 5 extra servers to deploy 5 replicas of the GTP
   d. Upgrade the GPU on the cloud server to make GTP run faster
   e. Move the GTP service to another cloud server located in US East
   f. Change the GTP model so that it takes less time to generate a token