Problem Set 1

15-440/15-640 Distributed Systems Spring 2016

Assigned: Thursday January 28, 2016

Due: Thursday February 4, 2016 (by the start of class)

Submission procedure:

- Create a .pdf of your answers and upload to Autolab**.**
- If you handwrite your answers, scan into .pdf before uploading.

Question 1 (48 points)

Bob wants to send a file of *F* bytes to Alice. The file is split into blocks of size exactly *M* bytes each. You can assume that *F* is a multiple of *M*, and that Alice knows these values in advance. The network between Bob and Alice has a latency of *L* milliseconds and a bandwidth of *B* bytes per second. Bob is considering two different approaches.

In the *Hare* approach, Bob sends each block as a UDP packet and only requires Alice to send a single ACK packet after all blocks have been received. Individual UDP packets are not acknowledged. After sending all the blocks, Bob waits for a time *T* to receive Alice's (single) ACK. If the ACK does not arrive within *T*, he repeats transmission of the entire file.

In the *Tortoise* approach, Bob again sends each block as a UDP packet. However, Alice must now ACK receipt of that packet. If no ACK arrives within time *T*, Bob retransmits that packet. He repeats this indefinitely, until the ACK arrives. He then proceeds to send the next packet, and so on until he receives the ACK for the last packet.

Bob is infinitely patient, and never gives up. For the different networking conditions described below, explore the merits of the *Hare* and *Tortoise* strategies for file sizes of 1 KB, 1 MB, and 1 GB. For all file sizes, the block size *M* is chosen so that exactly 10 packets are sent. As a simplifying assumption, you can assume that ACKs are ultra-reliable and therefore never lost. Clearly state and justify any other assumptions that you make, including how you choose a good value for *T*.

A. Suppose Alice and Bob live in different CMU dorms. The network between them is the CMU LAN. Latency *L* is one millisecond, and bandwidth *B* is one gigabit per second. The network is extremely reliable: the probability of a UDP packet being lost is only 10^{-10}. For different file sizes using the *Hare* and *Tortoise* strategies (i.e., total of 6 different combinations), how long is Bob's expected waiting time for successful file transmission?

B. Repeat (A) if the probability of UDP packet loss rises to 10^{-1} due to a flood in the Cyert Hall networking center.

C. Alice graduates from CMU and moves to Mars (the planet, not Mars, PA). Latency *L* is now 8 minutes (one way) and bandwidth *B* is one megabit per second. Fortunately, the network is still very reliable and the probability of UDP packet loss is still only 10^{-10}. For different file sizes using the *Hare* and *Tortoise* strategies, how long is Bob's expected waiting time for successful file transmission?

D. Suppose a solar storm temporarily increases the probability of UDP packet loss to 10^{-1}. Repeat (C) under these conditions.

E. Reflecting on your answers to A through D, crisply state the set of conditions under which the *Hare* strategy dominates the *Tortoise* strategy and vice versa.

Question 2 (16 points)

For each of the following use cases, identify the ***weakest*** RPC semantics that can be used (exactly-once, at-most-once, or at-least-once). In each case explain your answer.

A. Transacting a stock trade online
B. Checking on the status of a flight
C. Submitting a blog post that has already been composed in a local file and includes a timestamp and the user's name at the beginning of the blog text.
D. Creating a subdirectory in a distributed file system.

Question 3 (8 points)

On a lightly-loaded 10 Gbps LAN in a Facebook data center, end-to-end RPC communication between a client and a server for some operations is taking almost 100 milliseconds. None of the RPC operations involve large data transfers. Which of the following is the most plausible reason for this poor performance? Explain your answer.

A. Congestion-related queueing delays on the network.
B. Extremely inefficient packing and unpacking (i.e., marshalling and unmarshalling) of RPC packets.

C. Heavily loaded client, or server, or both.

Question 4 (8 points)

You have been hired by Uber as their lead architect for their driverless automobile system. Give one example each (with explanation) of a safety property and a liveness property in the system that you are designing.

Question 5 (8 points)

Excluding issues that arise from lack of a shared address space across machines, explain two different ways in which programming a distributed system using RPC is more complicated than programming using local procedure calls.

Question 6 (12 points)

To get around intellectual property laws that protect copyrighted material, the founder of EBooks_R_US has a novel business model. There is only one copy of each e-book legally acquired and paid for by the company. All of these e-books are located on a single server. If you `ssh` into the server, a user-friendly e-reader application can be used on any currently unused e-book. EBooks_R_US charges customers by how long they are logged in and using their e-reader. It is effectively "renting out" its e-books without making illegal copies of any e-book. The company does very well initially, with a rapidly growing customer base. Unfortunately, as the company expands, there is growing concern about the security risk of allowing each customer to `ssh` into the server. The founders' legal counsel assures them that a small variant of their current business model would also be lawful. In the new model, each e-reader would be wrapped in software that allows its operations to be invoked remotely. Instead of logging into the server, the user merely runs client software that remotely operates the e-reader. This is just a different way of "renting out" an e-book without copying it. You may assume that all client-server TCP connections are authenticated and encrypted.

You have been hired to rapidly implement the software needed to support the new model. Your business partner says you should use REST to implement the communication between server and client. Your Technical Advisor. in contrast, suggests that you use RPC to export e-reader functionality. Time is of the essence. The e-reader is completely implemented in C++. Would you recommend use of REST or RPC? Explain your answer.