

1. Consider the game described in Question 1 of Problem Set 2.

Recall that players can issue three commands (*without any arguments*):

- **rotate** - rotate the spaceship by 90 degrees clockwise.
- **move** - move the spaceship forward by one cell in the direction that the spaceship is facing, when possible.
- **fire** - shoot the laser immediately.

For each of the commands, **move**, **rotate**, and **fire**,

- explain if it is necessary to send the command using a reliable transport protocol,
  - and if so, is it possible to modify the command such that it need not be sent reliably? If your answer to the second question is yes, explain how the command can be modified.
2. Consider a client/server implementation of two player Pong game similar to your Assignment 2.

- Someone suggested that we apply redundant data bundling (RDB) for TCP to packets exchanged between the clients and the server.

Is this useful? If yes, describe the type of packets (its content and purposes) for which RDB is useful. Otherwise, argue why RDB is not necessary for your implementation of Pong.

- The networking library ENet provides optional reliability – when sending a packet, the application can indicate with a boolean flag whether the packet should be delivered reliably or not.

Someone suggested that we always turn the reliability flag off (i.e., send unreliably) when sending paddle position updates in Pong. Explain why this can lead to undesirable result and suggest how can you use the reliability flag wisely to avoid it.

(Note that sending all paddle position updates reliably is not a good solution).

3. Let  $d(x, y)$  be the average network latency between two hosts  $x$  and  $y$ . We assume that:

$$\text{if } d(x, y) \approx 0, \text{ then } d(x, z) \approx d(y, z)$$

That is, if two hosts are very close (in terms of latency) to each other, they are assumed to belong to the same network, and therefore their latency to a third host  $z$  is similar.

Based on the assumptions above, and without using network coordinates, explain how you can reduce the number of RTT measurements (i.e., the number of probes) a client has to send in the game server discovery protocol. Pay attention to the actions of, and the interactions among, the master server, the game servers, and the clients.