Assignment 1 CS454/654

1. Recall the discussion on asynchronous communication architectures that make use of a shared persistent data space. Suppose you are tasked to design an RPC system on top of this architecture that provides "at least once" semantics. Outline the basic design of your RPC system. How do you handle the case where the server is not present for the client's request and/or the client is not present for the server's reply?

2. TCP treats lost packets as signals for congestion. It increases and decreases its send rate in response to these end-to-end signals. Suppose that each intermediate router along the path can, much like with TCP flow control, send data back to the sender to indicate whether its queue is more than 90% full (1-bit of information, either >90% full or <= 90% full).
   a. How would you change the TCP congestion control algorithm to take advantage of this information? Remember a router, unlike end-hosts, typically has a single queue for all incoming packets. Include in your discussion the impact of other flows and route changes on the effectiveness of your algorithm.
   b. How would you change your algorithm if the routers can indicate up to 16-levels of fullness?
   c. What if only a fraction of the routers on the Internet supports this functionality? How would you change your algorithm to accommodate a partial deployment?

3. Exceptions provide a uniform mechanism to handle unexpected events.
   a. Explain in detail how you can modify RPC and RMI to support exceptions.
   b. Does it make sense to implement exceptions as a remote object? Consider the case where the exception is not caught and may be re-thrown across multiple RMI calls.

4. Consider a client-server application where the client makes a remote procedure call to the server. The link latency between the client and server is 10 ms each way, marshalling and unmarshalling of the parameters and return value take approximately 1 ms (i.e. 1 ms to marshal and another 1 ms to unmarshal), and the server takes 100 ms to process each request and can only process one request at a time.
   a. How long would it take to complete two back-to-back requests in a synchronous RPC system? How long would it take in an asynchronous RPC system? Assume the client is interested in the response and waits for the acceptance of the request before continuing.
   b. Imagine if the result of the first request determines which remote procedure to call in the second request. How would that change the completion time? What if the first request returns only true or false? Can you take advantage of this information to reduce the completion time, and does it require that the remote procedures have any special properties?

5. We talked about “at least once” and “at most once” RPC semantics in class. The most general definition of “at-most-once” semantics is very weak. I can claim that the RPC executed “at most once” by not sending a request at all. A stricter definition requires that an RPC that provides “at most once” semantics does not suffer from omission failures, that is, failures due the loss of request or result messages.
    a. Does a standard request-reply protocol that uses TCP as the transport layer provide the stricter definition of “at most once” semantics? If not, explain what additional mechanisms are required. Consider the case where connections are broken due to extended timeouts.
    b. There is also “exactly once” semantics which was not discussed in class where the RPC request is guaranteed to execute once and only once. What properties of the network, client, and server would be required to support “exactly once” semantics and how would you implement these properties? You can assume that, at some point in the future, the network, client and server will all be functional at the same time. However, do not assume that the network/client/server can never fail; they are prone to fail independently. How would providing these properties affect the performance of the system?

6. Textbook questions:
    a. Chapter 2 question 5 (HINT: Latency)
    b. Chapter 2 question 11
    c. Chapter 4 question 5