Homework 1

Due: October 5th, 2015 23:59:59pm

Please typewrite your answer to the homework problems, save your file in `.pdf` format, name your file as "your BU email id"-homework1.pdf (e.g., jdoe-homework1.pdf), and submit it to Blackboard before deadline. Late homework will not be accepted.

Problem 1. (10 pts) (Chapter 2 Problem 5 in Tbook)

In a structured overlay network, messages are routed according to the topology of the overlay. What is an important disadvantage of this approach? (Hint: latency.)

Problem 2. (15 pts) Consider a Chord ring based on 7-bit node identifiers and keys, nodes {15, 26, 47, 48, 51, 69, 75, 106, 110, 127} are present on the ring. Provide the finger tables for nodes 15, 106, 127.

Problem 3. (10 pts) List all hops for the following key lookups for the Chord system described in Problem 2:

source	key
15	12
26	3

Problem 4. (10 pts) (Chapter 2 Problem 11 in Tbook)

Consider a BitTorrent system in which each node has an outgoing link with a bandwidth capacity B$_{out}$ and an incoming link with bandwidth capacity B$_{in}$. Some of these nodes (called seeds) voluntarily offer files to be downloaded by others. What is the maximum download capacity of a BitTorrent client if we assume that it can contact at most one seed at a time?

Problem 5. (10 pts) (Chapter 4 Problem 6 in Tbook)

One way to handle parameter conversion in RPC systems is to have each machine send parameters in its native representation, with the other one doing the translation, if need be. The native system could be indicated by a code in the first byte. However, since locating the first byte in the first word is precisely the problem, can this actually work?

Problem 6. (10 pts) (Chapter 4 Problem 18 in Tbook)

Explain why transient synchronous communication has inherent scalability problems, and how these could be solved.

Problem 7. (10 pts) Consider a client-server application where the client makes remote procedure calls to the server. The link latency between the client and the server is 20 ms each way. Marshalling and un-marshalling of the parameters and return value take approximately 1 ms, respectively. The server takes 150 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?
(b) 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. (Figure 4-10 (b) in Tbook.)

Problem 8. (10 pts)

Traditional RPC mechanisms cannot handle pointers. What is the problem and how can it be addressed?

Problem 9. (15 pts) Consider the following scenarios for RPC operations in an asynchronous system, which of the three semantics {at least once, at most once, maybe} best describes the outcome in that scenario?

(a) The client keeps retransmitting the request until its receives a reply. Server caches all past requests it has received and replies it has sent. Upon receiving a request, it first checks to see if it has seen this request before and if the reply to this request is in the cache – if yes, it returns that reply otherwise it re-executes procedure.

(b) Same as (a) above, except that the server only caches requests and replies processed within the past 1 hour.

(c) The server does not keep a cache. It simply executes every request it receives. The client keeps retransmitting request until it receives a reply. In addition, the network might duplicate messages.

(d) Network is dropping all messages.

(e) Server has crashed, and will not recover.