15-440/15-640: Homework 1

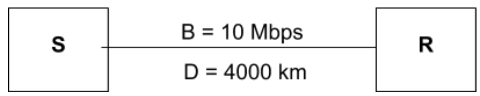
Due: September 23, 2018 11:59pm

Name: ______________________________

Andrew ID: ____________________________

1. Networking (20 Points)

Part A (10 points)



Consider the situation depicted in the figure above. The sender S want to send a file of size $S = 1Gbit$ to the receiver R over a direct link of bandwidth $B = 10Mbps$ and of length $D = 4000km$, where packets travel at $\frac{2}{3}$ the speed of light ($c = 3 \times 10^8 m/s$). The file is subdivided into packets of size $P = 5Kbit$ and a stop and wait protocol is used to ensure correct delivery.

(a) Suppose the client sends one packet at a time, then waits for an ACK before delivering another packet. How long will it take to transfer the whole file? What is the effective bandwidth? [**5pts**]

(b) To ensure optimal bandwidth usage, we have to make sure that the sender is never idle i.e. there are always packets being written. The sender is now allowed to transmit up to N unacked packets at a time, and then has to wait for an ACK before transmitting the next packet. Note that ACKs are for individual packets; they are not cumulative. What is the minimal value of N that ensures optimal bandwidth usage? [**5pts**]

Part B (10 Points)

(a) For each of the following scenarios, state whether you would use TCP or UDP. Briefly explain your reasoning. [**1pt each**]

  i. A high performance video streaming application.
  ii. A banking application
  iii. An online video game.
  iv. Server-residing cache using broadcast invalidations (server broadcasts message to multiple clients that their cache is invalid for a particular key).

(b) State and explain one advantage and one disadvantage of the "IP waist" in the hourglass model. [**3pts**]

(c) We saw in lecture that web page retrieval involves several steps, which could potentially blow up the latency. What solution is employed to deal with this issue? Briefly explain. [**3pts**]

2. # Concurrency (15 Points)

Part A (6 Points)

```
 1.   C.Init():
 2.     C.mutex = NewMutex()
 3.     C.readCvar = NewCond(C.mutex)
 4.     C.writeCvar = NewCond(C.mutex)
 5.     C.value = undefined
 6.     C.full = false
 7.     return
 8.
 9.   C.send(value):
10.     C.mutex.lock()
11.     If C.full {
12.       C.writeCvar.wait()
13.     }
14.     C.full = true;
15.     C.value = value;
16.     C.readCvar.signal()
17.     mutex.unlock()
18.
19.   C.receive(value):
20.     C.mutex.lock()
21.     If (not C.full) {
22.       C.readCvar.wait()
23.     }
24.     C.full = false
25.     value = C.value
26.     C.writeCvar.signal()
27.     mutex.unlock()
```

Identify the bugs in this code. For each of them clearly state the line number, the reason why it is a bug, and how you would fix it (*Hint: some values are being overwritten, while others are being read multiple times*)

Part B (9 Points)

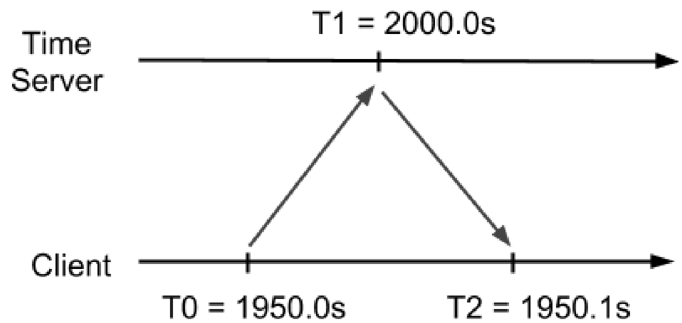
(a) Explain the main benefit of using condition variables instead of spin locks. [**4pts**]

(b) What is the main difference between a livelock and a deadlock. [**2pts**]

(c) Consider a Producer/Consumer pattern with is working on a fixed number of data items. Do you anticipate that a correct implementation uses semaphores or mutexes? [**3pts**]

3. # Time & Synchronization (25 Points)

Part A (9 Points)



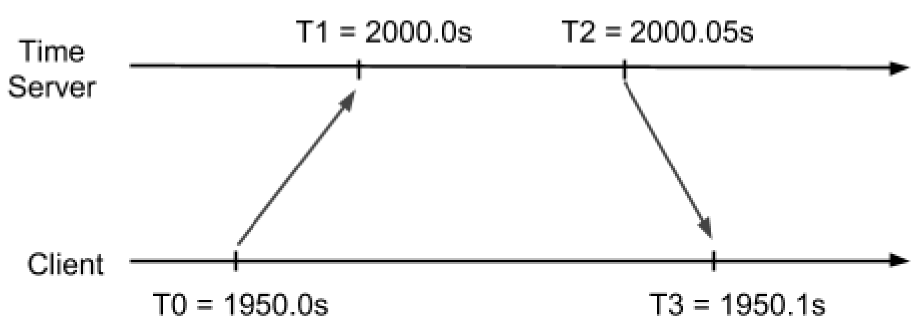
Consider the scenario depicted above. The system guarantees that the minimum one way delay between the client and the server is 25ms.

(a) How will the client update its local time according to Cristian's algorithm? **[1pt]**

(b) Keeping $T0$ and $T2$ constant, what is the best case error of Cristian's algorithm? When does this situation arise? **[1pt]**

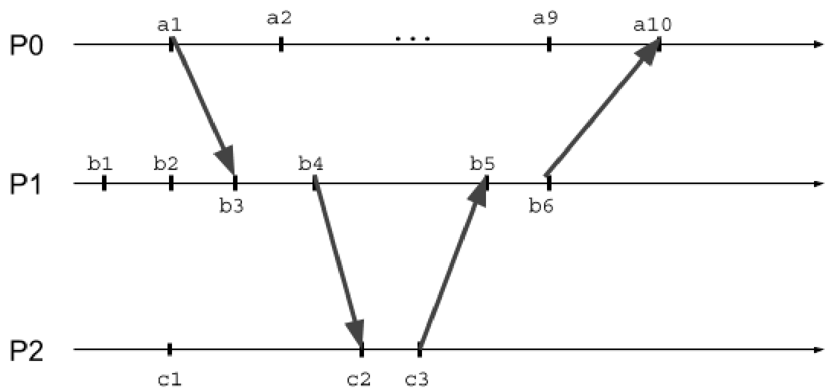
(c) Keeping $T0$ and $T2$ constant, what is the worst case error of Cristian's algorithm? When does this situation arise? **[2pts]**

(d) In a more realistic setting, the time server would not be able to immediately respond to the client's request. We therefore have the setting below. Determine the estimated network RTT and time offset. **[3pts]**



(e) How would you implement the function that updates the local time on the client? Describe your reason in 2 sentences. **[2pts]**

Part B (16 Points)



(a) Consider the situation depicted above. Events a2, ..., a9 are all local to the process P0. Fill in the following table using the various algorithms described in class. Assume that P0's id is 0, P1's is 1 and P2's id is 2. **[14pts]**

Event	Lamport's Algorithm	Totally Ordered Lamport Clock	Vector Clock Algorithm
a1			
a2			
a9			
a10			
b1			
b2			
b3			
b4			
b5			
b6			
c1			
c2			
c3			

(b) What are the advantages and disadvantages of using Lamport Clocks instead of Vector Clocks? State at least one advantage and one disadvantage. **[2pts]**

4. # Remote Procedure Calls (15 Points)

Part A (10 Points)

Consider the following code snippet.

```
struct Node {
  int value;
  Node* next;
}

void processList(Node *head);
```

The processList function is very computation heavy, so clients delegate the work to a more powerful backend server using an RPC.

(a) When implemented as a local procedure, processList only needs to copy an 8 bytes pointer to the head of the list. How will this change when it comes to implementing the function as an remote procedure? What drawbacks can arise? [**5pts**]

(b) Provide a high level description of how you would marshall this list on the client, and of how you would unmarshall it on the server. [**5pts**]

Part B (5 Points)

For each of the following scenarios, state whether you would use an at-most-once semantics or an at-least-once semantics. Explain in no more than 2 sentences. [**1pt for each**]

(a) Booking a flight from a travel agency

(b) Retrieving the value associated with a key

(c) Withdrawing cash from bank account

(d) Sending a message/post on social media

(e) Registering for a class at CMU

5. **Distributed File Systems (25 Points)**

(a) Briefly state 2 advantages of using NFS over AFS. [**2pts**]

(b) Briefly state 2 advantages of using AFS over NFS. [**2pts**]

(c) For each of the following scenarios, choose whether a file system like Coda, a file system like LBFS, or neither would be a better fit. Briefly justify your decision. [**1pt for each**]

   i. An application that allows multiple users to collaborate on a single document.

   ii. Operating on a network with frequent disconnects

   iii. Limited network resources (High latency)

   iv. An application where, overall, the data being sent has a lot of redundancy.

(d) Briefly explain the CAP theorem (2 sentences, max). [**3pts**]

(e) You are put in charge of designing an interplanetary distributed file system meant to be used on Earth and Mars. The clients on Mars are rovers that execute software from the file system to decide where to go. The rovers will take pictures and gather data at regular intervals and upload these to the file system as well. The software on the file system will constantly be updated by developers on Earth and the rovers will apply these updates whenever they come. However, compiling and loading the new software takes several hours and the rover cannot move or gather any data during this time.

   i. In case of a network partition, is 'consistency' or 'availability' a more desirable property of this file system? Briefly explain your decision. [**3pts**]

   ii. Based on your answer from (i.) would optimistic replication or pessimistic replication be a better choice for this file system? [**2pts**]

   iii. Would you implement client-side caching for this file system? Briefly justify your decision. [**4pts**]

   iv. It turns out a lot of the images on Mars are very similar. How can this fact be used to optimize the file system? Briefly explain your answer. [**5pts**]