NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER ASSESSMENT FOR
Semester 2 AY2014/2015

CS4344 Networked and Mobile Gaming

April 2015 Time Allowed 2 hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains FIVE (5) questions and comprises FOUR (4) printed pages, including this page.
2. Answer **ALL** questions. State your assumptions, if any, clearly.
3. The total marks for this paper is A HUNDRED (100).
4. Answer all questions in the answer book provided.
5. This is an **OPEN BOOK** assessment.

1. (30 points) **Response Time and Consistency**

   We define the *response time* of an action in a particular game implementation as the time that elapses between player issuing an action (e.g., moving the mouse) and the effect of the action appearing on screen (e.g., pong paddle starts to move) and *acceptable response time* of an action as the threshold above which the performance of the player is impaired by the response time. In this question, we want to relate an in-game action with the acceptable response time, based on the category of the action.

   Player actions in a game can be classified based on two attributes: *precision* and *deadline*. Precision measures the accuracy required to complete an action successfully; Deadline measures the time by which the action must be completed to achieve the intended outcome.

   Based on these two attributes, we can classify game actions loosely into four categories:

   - **HP-TD**: high precision with tight deadline
   - **LP-TD**: low precision with tight deadline
   - **HP-LD**: high precision with loose deadline
   - **LP-LD**: low precision with loose deadline

   Example of a **HP-TD** action: consider the action of shooting at a distant enemy in a first person shooting game using a sniper weapon. The player would need to aim accurately. The further away the enemy is, the higher the required precision of shooting action is (because far-away enemy appears small on the screen and the weapon has to be aimed exactly at the enemy). Furthermore, the faster the enemy moves, the tighter the deadline of this shooting action is (because if the enemy moves outside the range, the action of shooting would not achieve its outcome).

   Example of a **LP-LD** action: consider the action of exploring the game world during peacetime in a real-time strategy game. The player orders a unit to move to a spot on the map in order to uncover the fog of war. Such exploring action does not require a high precision, as the goal of exploration would still be completed even if the location of the destination is slightly different. Furthermore, exploring has a loose deadline – even though it is advantageous for a player to discover its surroundings as early as possible, there is no urgency for a player to do so.

   (a) (5 points) Give an example of a **HP-LD** game action. Explain your answer.

   (b) (5 points) Give an example of a **LP-TD** game action. Explain your answer.

   (c) (5 points) Explain the relationship between the precision of an action and its acceptable response time.

   (d) (5 points) Explain the relationship between the deadline of an action and its acceptable response time.

   (e) (10 points) We often trade off between the response time and consistency when designing multi-player games. Short-circuiting is one such mechanism that sacrifices the state consistency among the players to lower the response time.

Out of the four categories of actions above, which are the ones (if any) that can lead to better playing experience if we reduce the response time but increase the states inconsistency as a result? Explain your answer.

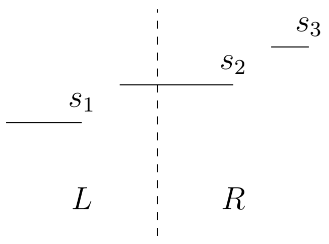
2. (25 points) **Generalized Interest Management**

Consider generalized interest management schemes, in which each player defines a *subscription region* and an *update region* in a k-dimensional attribute space. The server maintains the list of all subscription region $S = \{s_1, s_2, ...\}$ and all update regions $U = \{u_1, u_2, ...\}$ and determines which update region overlaps with which subscription region, in order to decide whether to forward a message to a player. Recall that the sort-based DDM method computes the overlap for each dimension separately. Assume that, for each dimension d, there is a function call DDM(S, U, M, d) that takes in the two sets of regions and updates a matrix M with $M(i, j)$ set to 1 if and only if s_i overlaps with u_j in the d-th dimension.

Suppose that, for each dimension, we split the attribute space into two equal halves, denoted L and R. We classify the subscription regions and update regions into three categories:

- S_L and U_L: subscription and update regions that are contained within L;
- S_R and U_R: subscription and update regions that are contained within R;
- S_{LR} and U_{LR}: subscription and update regions that span across L and R.

The figure shows an example. In this example, s_1 belongs to S_L, s_2 belongs to S_{LR}, and s_3 belongs to S_R.



This categorization produces nine possible pairings between a set of subscription regions and a set of update regions.

(a) (5 points) Among the nine pairs above, list down the pair (or pairs) where subscription region in one set *never* overlaps with an update region in another set.

(b) (5 points) Among the nine pairs above, list down the pair (or pairs) where any subscription region in one set *always* overlaps with an update region in another set.

(c) (15 points) Based on your answers above, sketch a recursive implementation of DDM(). You can assume the existence of a function Split(X) which takes in a set of regions, and partition the regions into three sets X_L, X_R, and X_{LR} as above by diving the attribute space of X along a dimension into two halves.

3. (15 points) **TCP**

   (a) (8 points) For each of the techniques to reduce TCP's latency below, indicate if the technique (I) can be implemented with changes to TCP on the sender-side only, (II) can be implemented with changes to the TCP stack on the receiver-side only, or (III) requires changes to the TCP on both the sender and the receiver side.

   Briefly (in no more than one sentence) explain your answer.

   i. (2 points) turning off delayed acknowledgment

   ii. (2 points) triggering fast retransmission after only one duplicate ACK

   iii. (2 points) replacing exponential backoff with linear backoff

   iv. (2 points) redundant data bundling

   (b) (7 points) One technique to save power in a mobile multiplayer game is to put the wireless network interface of the client into sleep mode for as long as possible. During sleep mode, the client cannot send nor receive any message. You have seen how one can predict how long we should put the network interface to sleep without having an impact on the state consistency of the games.

   Putting the network interface into sleep mode for too long, however, can reduce the throughput of TCP and increase TCP's delay between the client and the server. Explain why it might be so.

4. (15 points) **Matchmaking**

   The matchmaking process aims to group players waiting in the game lobby into game sessions such that the maximum latency between any two players is low. Both hierarchical clustering and QT clustering method for matchmaking require every player to measure the RTT to every other player. Suppose we have n players waiting in the lobby. A total of $n(n-1)$ probe messages will be generated to measure RTTs among these players. Even if we assume that RTT is symmetrical, we will still have $n(n-1)/2$ messages.

   Using virtual network coordinates, sketch a method in which the server can perform matchmaking without requiring every player to measure the RTT to every other player.

5. (15 points) **VON**

   Voronoi Overlay Network, or VON, is a useful technique for interest management in a distributed multi-player game without a centralized server. In the design of VON, when a node moves, position updates are sent to all connected neighbors of the node, i.e., the AOI neighbors plus any enclosing neighbors beyond the AOI.

   Argue why it is not always necessary for a node to send position updates to the enclosing neighbors that are beyond the AOI of the node, and it suffices in this case for the node to maintain a connection to these enclosing neighbors. Explain clearly the condition under which the node needs to send position updates to an enclosing neighbor, and the condition under which position updates need not be sent.

END OF PAPER