National University of Singapore
School of Computing

CS4344 **Problem Set 3: Hybrid Architecture** Semester 2 14/15

1. To support seamless game world in an MMORPG game, the following approach, called replicated zoned servers, is proposed. The game world is organized into regions and the regions are mapped onto servers. Each server is responsible for maintaining the primary copy of states within a region, as well as replica of states in neighboring regions (two regions are neighbors if they share a border). Note that regions are not necessary contiguous.
   (a) What are the pros and cons of this approach, compared to mirrored servers architecture and zoned servers architecture?
   (b) Describe how states should be synchronized among the clients and servers using the replicated zoned servers architecture.

2. Consider the following network architecture for an MMORPG. The system consists of many servers, organized into a k-ary tree of height n. A player/client can connect to one or more servers in the tree (See Figure 1). A host can only communicate directly with other hosts it is connected to.

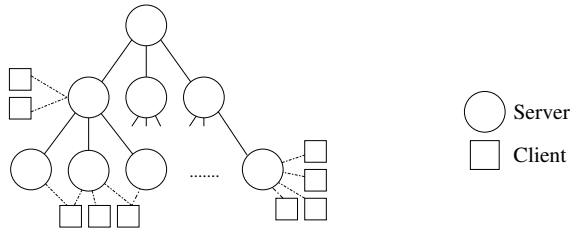

Figure 1: An example depicting a system of servers organized as a tree with $k = 3$ and $n = 3$.

The game world is partitioned into uniformly size rectangular cells. Each server is responsible for maintaining the states of one or more cells. The states of a cell is replicated among n servers, forming a path from the root to a leaf server in the tree. The root server maintains the states of the whole game world (i.e., all cells). The set of cells maintained by a server is always a subset of the cells maintained by its parent.

Answer the following questions. Note that the questions are inter-related, so you should read *all* the questions and think carefully about the answers before you answer anyone of them.

   (a) After partitioning the world into cells, the system needs to decide which cells should be maintained by which server. Give two desirable properties of the set of cells assigned to a server. Why are the property desirable?
   (b) When a player joins the game, his/her game client needs to connect to one or more servers. Give two factors that should be considered in deciding which server or servers a client should connect to? Justify your answer.
   (c) Give a game scenario (e.g. capability or action of an avatar) where it is advantageous for a client to connect to a non-leaf server.
   (d) Suppose a player updates its own state, causing the game client to send an update message. Which other servers/clients should eventually receive this update message?
   (e) Is it better to organize the servers into a short and fat tree (large k, small n) or a tall and thin tree (small k, large n)? Justify your answer.
   (f) Someone suggested that we should add connections among servers at the same level in the tree. Will this improve a player's gaming experience? Why? or Why not?

3. Consider a real-time strategy game that supports 32 players, implemented as a point-to-point game using stop-and-wait bucket synchronization. As every player needs to send an update to every other player in every round, a heavy message overhead is incurred – about 1000 messages are exchanged every round!

   In an effort to reduce the number of messages, the following communication architecture is proposed:

   - Group the players into four groups, each with 8 players.
   - Players within a group communicate in a point-to-point manner.
   - One of the players in each group is designated as the group *leader*.
   - The four group leaders also communicate in a point-to-point manner. During a round, each leader consolidates the eight event updates from its group into one message and sends it to the other three leaders.
   - Each leader then forwards the consolidated event updates from the other three leaders to players in its group.
   - Event updates sent during a round are put into a bucket. After a predetermined lag, when all 32 event updates from all players are available, a player processes the event updates and updates the states.

   Figure 2 below illustrates the communication structure among the players.

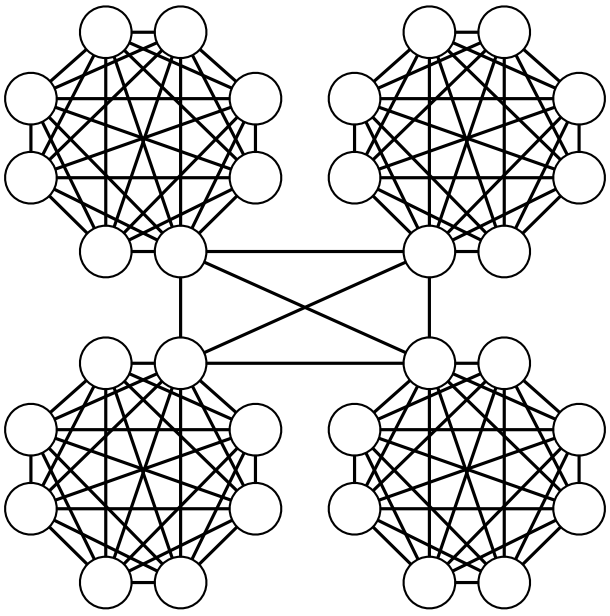

Figure 2: The proposed communication structure for the 32-player RTS game.

(a) Estimate the number of messages exchanged in each round using the proposed scheme.

(b) One of the key parameters of bucket synchronization is lag. Explain how you can determine the appropriate value of lag in the proposed scheme.

(c) Sketch a method to group the players into four groups of eight each and to designate one player from each group as a leader, such that the lag is reduced.