Patrick Trinkle

CMSC621 Fall 2009

Homework 2

### Exercise 6.13

A distributed system may have multiple, independent resources. Imagine that process 0 wants to access resource A and process 1 wants to access resource B. Can Ricart and Agrawala's algorithm lead to deadlocks? Explain your answer.

Ricart and Agrawala's algorithm defines that all requests for a resource must be approved by all nodes. If a node does not receive a reply from all systems it does not attempt to access the resource. The algorithm does not detail whether or not a process can hold a resource X, and then request resource Y. Therefore, if process 0 holds B and requests A while process 1 holds A and tries to access resource B there can be a deadlock. However, this is dependent on the system configuration and not strictly specified by the algorithm.

The algorithm does not strictly prevent deadlocks. It also doesn't lead to them necessarily.

### Exercise 6.14

Support that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the bully algorithm. What happens?

Two possible situations:

1. **Process 0 has a higher ID.**

* Process 0 and Process 1 both ask nodes with higher IDs if they would like to be leader. If Process 0 hears back from a node 3; 3 takes over. If Process 1 also hears back from a node (4). Then 3 and 4 will broadcast success, at which point the higher of the two will restart the election.
* Process 0 and Process 1 both ask nodes with higher IDs if they would like to be leader. If Process 0 hears nothing and Process 1 hears nothing; they both assume they're leaders and broadcast their victories. When Process 0 receives the victory from Process 1 it will restart the election because its ID is higher.

1. **Process 0 has a lower ID.**
   * Process 0 and Process 1 both ask nodes with higher IDs if they would like to be leader. If Process 0 hears back from a node 3; 3 takes over. If Process 1 also hears back from a node (4). Then 3 and 4 will broadcast success, at which point the higher of the two will restart the election.
   * Process 0 and Process 1 both ask nodes with higher IDs if they would like to be leader. If Process 0 hears nothing and Process 1 hears nothing; they both assume they're leaders and broadcast their victories. When Process 1 receives the victory from Process 0 it will restart the election because its ID is higher.

Effectively, regardless of how many elections are held simultaneously if a victory broadcast is received from a lower process ID a new election is held.