538B Assignment 3 Write Up - Stewart Grant

Design

My design has 3 components. Failure detection, Critical Section Acquisition, and Joining.
Failure detection: I designed my system to detect failure first. The specification noted that "Every non-failed node can communicate with every other node" and "The inter-node round-trip time is upper bounded by a constant RTT and can be used to detect node failures". These two rules allow for easy failure detection. My nodes broadcast a heartbeat messages at a frequency five times less than the round trip time. If a node does not respond it is dead. Because all nodes can communicate, all nodes will know that a node is dead within the heartbeat time. Any higher level function that requires consensus such as Critical Section Acquisition, and Joining are can be safely retried after a time of 2*heartbeat under the assumption that every other node has learned of the failure.
Critical Section Acquisition: The critical section is obtained by performing the ricart-agrawala algorithm. A node requests the critical section via broadcast, after receiving acks from all other nodes it executes the critical section. Jobs to be done in the critical section are stored in a queue. When a node enters the critical section it pop's the queue and executes the job. Nodes with non-zero queues will make a critical section request.
Joining: To join the cluster a joining node sends a request to a node in the cluster. The joining node only communicates to a single node in the cluster, until it has been accepted. If another node requests to join with a node that is not yet in a cluster, the node not yet in the cluster will be silent, and the joiner will assume they are dead. Joining nodes send heartbeats to the node it requested the join from, and exists if the node in the cluster failed. Upon receiving a join request a node enqueues a join job to its critical section queue. Once a joiner enters the critical section it performs a two phase commit with the other nodes in the cluster. Upon success the joiner sends it's set of peers to the joining node. Upon receipt of the peer set a node is in the cluster. This joining strategy has the effect that node joins are serialized. If nodes fail during 2PC the joiner will wait for the retry time, and then try again. This strategy allows for many nodes to concurrently join the cluster with knowledge of any node in the cluster.

Dinv

I maintained critical section state by having a boolean named critical. I logged the value of the variable at two points. One inside the critical section, and one outside. I used dump annotations to get a precise view of the state. I was able to detect the safety property a couple of times, but I was not thinking, and made my program really threaded which really threw Dinv off. Because I could not detect it constantly on all runs, I decided to throw in the towel. In the cases where I was able to detect the property the output for each node in its critical section shows all other

nodes to not be in the critical section. In output which did not detect the safety property, the critical variable was always false. In both cases the critical section is safe, but