Trav Feller
Lab 9

1. Did you use a single message identifier for all message types in your system, or different one for each type of message? Why?
    1. I did not have to use a message identifier, I just used differing rules. For rumors, I had rumor rules. For seen messages, I used seen rules. For threshold seen messages, I had rules. This made it simple, I always knew what my message content contained.
2. Did you have to change how your seen messages worked? Why or why not?
    1. Not really, I just added an option to my heartbeat that sent a threshold seen message, and a rule that selected on the event raised by a peer node. This just made sure that my current receiving node updated its ledger to reflect the missing values sent by the peer.
3. How did the state-oriented CRDT we used for Lab 9 differ from the operation-oriented CRDT we used in this lab?
    1. The CRDT from the last lab was only state-based, meaning it was keeping track of the exact values of all the nodes and their ledgers. This current lab is keeping track of changes or operations taken to each node. In this case, you will not be able to easily reverse the process to see what the state was for a node in a specific moment of time.
4. Is it possible for a node to issue two positive threshold violation messages (i.e. value = 1) without an intervening negative threshold violation messages (i.e. value = -1)? Justify your analysis. What are the consequences of such a scenario?
    1. No. Not with my setup because my setup adds a value to the ledger each time the temp is updated so it would have to put a -1 value in before the two 1's. If this did slip by, it would be by computer error or a sneaky bug in my code, and this would forever set off my total value. If this was a recurring issue, my total violations value to be way off and thus useless.
5. How does gossip messaging combined with CRDT compare with Paxos? Consider the threshold counter we implemented for this lab. How would it be different if you tried to use Paxos to implement it?
    1. Paxos works by asking at least 50% + 1 nodes in your distributed system if a value has been accepted or rejected and majority wins. In our case, each of the violation counters will eventually reach the same value, but until they do there will be discrepancy, whereas paxos just sends the same thing to over half the nodes and waits for a common response.
6. How does gossip messaging combined with CRDT compare with Byzantine consensus (like in a blockchain)?

1. This is very similar to before where the same request is sent out to all nodes and you have to wait till 2m+1 nodes respond where m is the number of nodes that you can be willing to return incorrect data. This way, if some nodes are faulty, you will still have a majority of nodes that are giving correct information.

Hint: for the last two questions, consider whether consensus, consistency, and convergence are the same thing. Also, consider the goals of different algorithms.