

% of A1 mark	Difficulty	Goal of testing	Scenario (specific values are approximate)
3 Mon	Simplest	Initialize can't be invoked twice	Call initialize twice, ensure second time returns an error
6 Mon	Simple	AddMonitor: check that node is able to determine definite failures.	Have the student node monitor a node which won't respond to messages. Set a low lost_msgs_thresh and expect failure within ~30s.
5 Resp	Simple	StartResponding, StopResponding: check that node is able to respond to heartbeats and is able to properly stop responding.	Have the student code start responding, ensure it responds correctly for a few messages. Ensure we get no failure notifications. Then have the student node stop responding and ensure now we receive failure notification about the node).
6 Mon	Simple	AddMonitor: check that node is able to properly monitor another node.	Have the student monitor a server which will respond to messages. Set a higher lost_msgs_thresh(~100), run for a few minutes and make sure no failures are reported.
10 Mon	Medium	AddMonitor, RemoveMonitor: Check that the node is able to determine failures that happen after sometime and that RemoveMonitor works correctly.	Have the student monitor a server which will respond to messages. Have the server stop responding after some time elapsed, expect failure detected within ~30s. Ensure there is not more than one failure notification (within ~30s).
10 Mon	Medium	AddMonitor: Check that the RTT is updated correctly, if the server is responding slower.	Have the student monitor a server which will respond to messages. Set a higher lost_msgs_thresh(~50). Have server respond during a fixed interval (i.e., one ACK every second). Run for a while and make sure no failures are reported.
5 Resp 10 Mon 15 Total	Medium	AddMonitor, StartResponding: check that the node is able to send heartbeats and monitor a node at the same time.	Call start responding on student node. Setup a node and have it start responding. Have student node monitor that node, ensure no failures are reported. Have that node monitor student node, ensure no failures are reported
10 Mon	Medium	AddMonitor: check that the node can monitor multiple nodes at the same time.	Setup multiple nodes (5) and have them start responding. Have student node monitor those 5 nodes with high threshold, ensure no failures are reported
5 Resp	Medium	StartResponding: check that the node can be monitored by more than one node simultaneously.	Call start responding on student node. Have multiple nodes(5?) monitor student nodes for 30s with high threshold. Check that no failures are reported.
15 Resp 15 Mon 30 Total	Medium Hard	AddMonitor: check that node is able to send heartbeats to 3 nodes and monitor 3 nodes at the same time.	(harder version of the 2nd case) Call start responding on student node. Setup 3 nodes and have them start responding. Have student node monitor those 3 nodes, ensure no failures are reported. Have those 3 nodes monitor student node, ensure no failures are reported
30 Resp 70 Mon 100 Total	*	*	*

Notes:

- In general, our testing scenarios will not be focused on testing corner-cases in the A1 specification. For example:
 - We will not test multiple invocations of AddMonitor with the same LocalIP:LocalPort arguments in the same session.
 - We will not test multiple invocations of AddMonitor with the same RemoteIP:RemotePort arguments in the same session.