Machine Problem 2 Report

1. Design description

In our design, we use a simplified SWIM-like algorithm to realize the maintenance of distributed membership group and failure detection, including joining, leaving, pinging and detecting failure due to interrupt. Due to the condition that only 3 machines would fail at the same time, every machine in our system would monitor 3 other machines (3 after itself in the total list of machines in group). Every machine has 2 lists to maintain, one is the monitor membership list (3 machine to monitor), other one is the total member list (including all nodes in the group). With the distributed structure of MP1, we can debug our MP2 method very quickly and see the log information while sending command between different machines.

1) **Join:** VM1 is set to be our introducer machine and would run first, and join itself in group first. Then other new machine which is going to join the group would send the join request to introducer first, and then introducer would update its total member list and its monitor membership list, then broadcast its updated total member list to the other member in the group. Other machine in the group including the new one received the updated list would update its own total member list and its monitor member list.
2) **Leave:** When a machine in group wants to leave the group voluntarily, it would first broadcast its node information (IP, ID, port number) to the machines in its monitor membership list. Then it would leave the group and stop receiving and processing other message coming from the members in the group. Once other members in its monitor membership list received the leave message, they would first detect whether the machine is in their total member list and monitor membership list. If so, they would delete the machine in both list, and broadcast the leave message to the other machines in their own monitor membership list.
3) **Ping and detect failure:** Every machine in group would ping the machines in their monitor membership list per second. If it does not receive the package back from the machine they ping on time, it would consider the machine it pinged failed, then it would delete the machine in its monitor membership list and total member list and renew its monitor membership list, and send the fail message to all the machines in its new monitor membership list. Every machine in the group would only delete the failed node once and broadcast once in the procedure.

2. Question

(i) The background bandwidth usage for 4 machines

When no membership changes, the machines only need to ping 3 other machines and send back ack package when pinged. In our method, every machine ping 1 machine in monitor membership list per second, so there would be 4 ping packages and 4 ack packages sent in a second. So the background

bandwidth usage for 4 machines would be **288 Bps.**

(ii) The average bandwidth usage whenever a node joins, leaves or fails (3 different numbers)

- Average bandwidth usage when a node joins: **721Bps**

When a node joins, it would send a message to introducer, and introducer would send the total member list to all nodes in group. So the bandage is different when there is 1 or 2 or 3 nodes in group before the other nodes join. So average nodes join bandwidth would be the average of the situations above.

- Average bandwidth usage when a node leaves: **648Bps**

A node leaving needs to send the leave message to the nodes in monitor membership list, and traverse in group. So when 4 nodes in group, 1 leaves, the leave message would be send 9 times. (9*72 = 648)

- Average bandwidth usage when a node fails: **438Bps**

A node failing would not sending message, and the rest of nodes would sent on its fail information in the group, which is 6 times when the group was initially 4 and 1 failed.

(iii) False positive rate

The data in the table records the first time of false positive appears. In the following table plot we can see that the higher message loss rate is, the less time to generate false positive is, and the false positive rate would be higher as well. Because we do not implement the indirect request in SWIM-like failure detection method. In our method, once the machine do not receive the ack package, it delete the node right away without sending the suspicious message first.

	Message loss rate	Average	Standard deviation	Confidence intervals
N=2	3%	24.6	15.93	[10.64, 38.56]
	10%	9.6	2.88	[7.07,12.13]
	30%	4.4	1.52	[3.07,5.73]
N=4	3%	18.2	8.56	[10.7, 25.7]
	10%	10.2	1.64	[8.76, 11.64]
	30%	2.4	0.55	[1.92, 2.88]