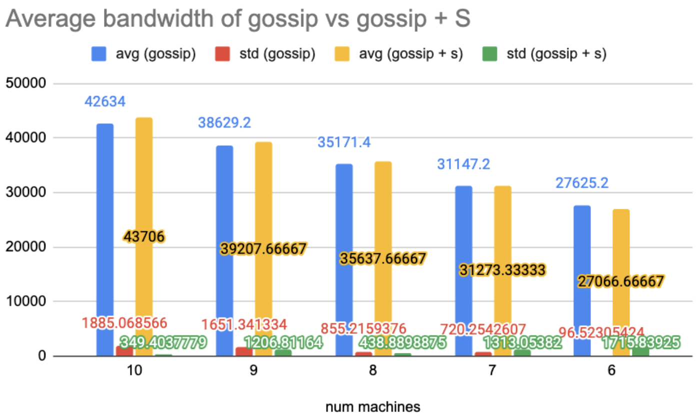
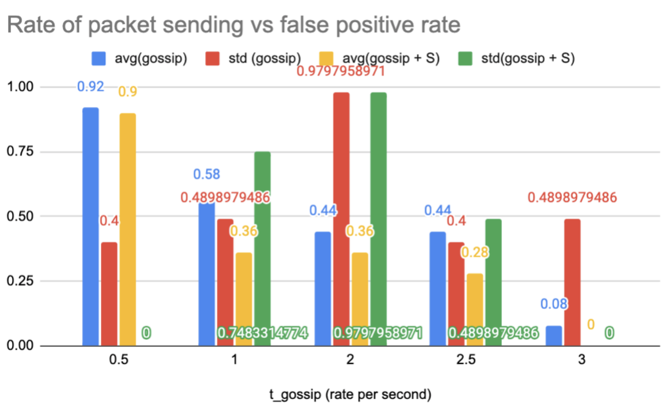
1.a) the bandwidth between Gossip vs. Gossip+S, in the no-failure scenario



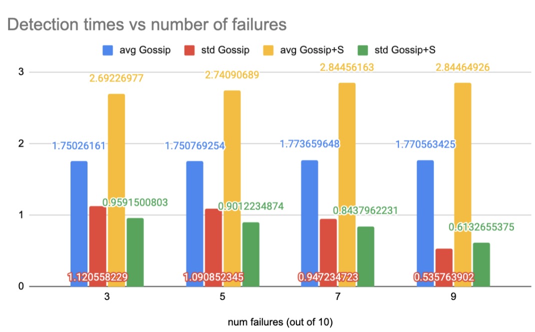
For this experiment, we varied the number of machines and kept all other variables constant (fanout, T_gossip, timeout). As we can see from the graph, a decrease in the machine count decreases the bandwidth (Bytes/sec) per machine. This is due to less nodes pushing data in the network and less nodes being represented in the membership list.

1.b) false positive rate when there are no failures



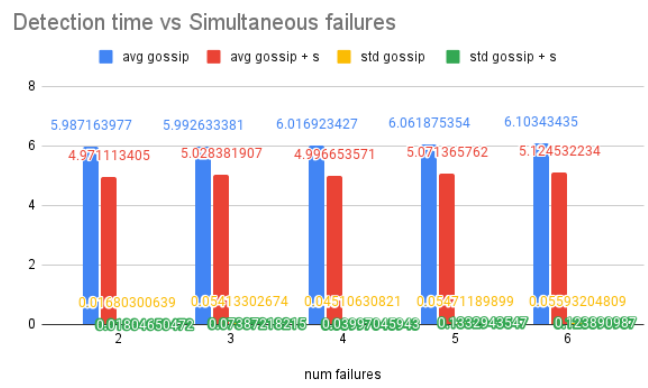
For this experiment, we kept the message drop rate at 0.2, number of vms at 5, fanout at 2, while varying the T_gossip (number of messages sent per second). This induces the higher average false positive rate as we decrease the number of times sent per second, considering the higher bandwidth due to constant time spent sending, but more network traffic.

1.c) Detection times as a function of the number of failures.



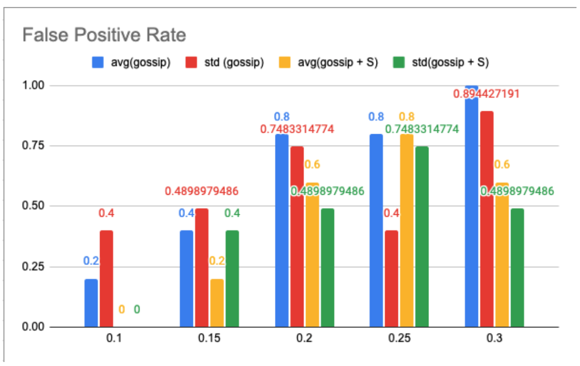
For this experiment, we varied the number of vms we ran and triggered a failure at a random node. As the number of vms decreases, the bandwidth decreases because less nodes are pushing information in the network over a set period of time. On average, the gossip+S takes more time, most likely because we allowed more time for nodes to evade their suspected down. Additionally, the average detection time decreases proportionally to node count, potentially because of the added bandwidth in the network, resulting in more data sent between nodes, resulting in the quicker detection of failures.

2.a) the detection time for failures, as a function of number of simultaneous failures,



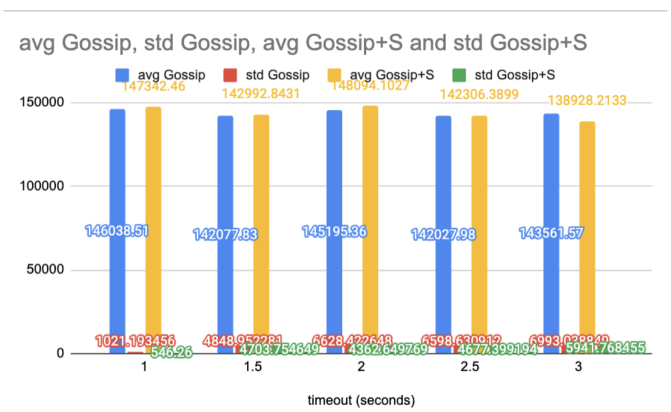
In this case, we kept all variables constant, but varied the number of vms that failed simultaneously. We then measured the average time for a failure to be detected by averaging first detection times of each node failure. We can see that the detection time for gossip+suspicion is higher than that of gossip, similar to 1c. With more simultaneous failures, the average detection time seems to increase, most likely due to network congestion.

2.b) false positive rate when there are no failures



This experiment consisted of letting the system run for 3 seconds under the provided conditions, and repeating that process 5 times to get the average and std of number of failures. Also, to induce false positives, our sending rate on each machine was two messages every second. As the drop rate increases, the failure rate increases, which seems logical. On average, the suspicion mechanism triggers less false positives than gossip, likely due to the higher time to allow nodes to "recover", thus the bandwidth remains constant.

2.c) Give the bandwidth usages of Gossip and Gossip+S



This experiment was conducted by measuring the average bandwidth of each node over the course of five seconds, with all factors except the timeout remaining constant. As we can see from the graph, as the timeout increases, the bandwidth of both Gossip and Gossip + S stays roughly the same. This is because while the fanout is fixed and less than the number of alive nodes, the bandwidth will remain the same as more messages will be sent to the remaining N-1 alive nodes. Additionally, the bandwidth of Gossip and Gossip + S stays roughly the same because regardless of the mode selection, the data held by the membership list remains unchanged.