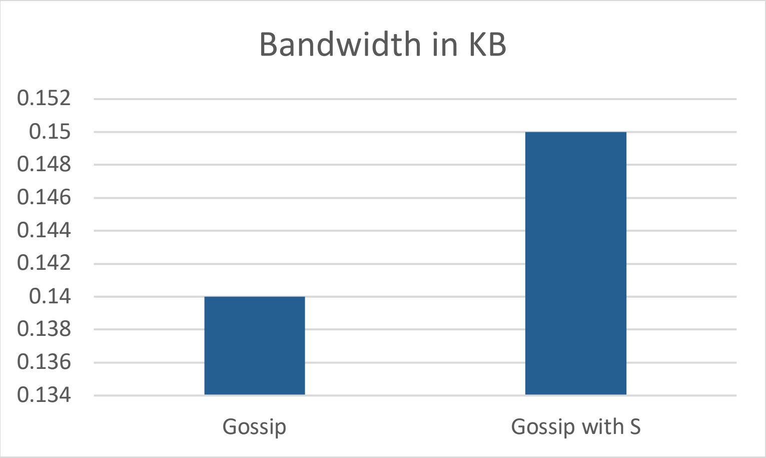
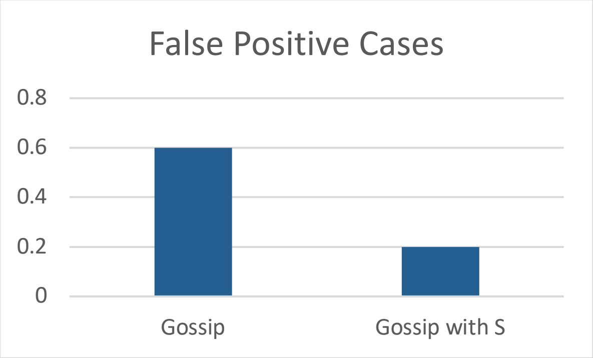
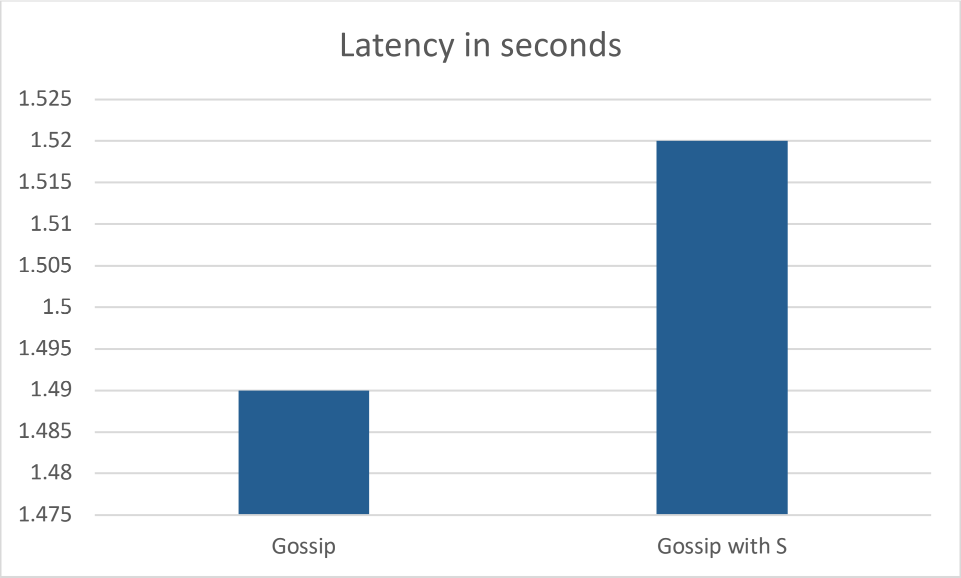
Part 1:



Having set T_suspect to 700 ms, and T_fail being 1000 ms, with T_gossip as 500 ms, we expect to see a high number of suspicion cases that require additional bandwidth to be resolved. The primary source is bandwidth is the regular gossip messages set to the T_gossip heartbeat interval, which is a summation of both RX and TX bandwidth on the UDP client and server socket. Gossip with suspicion is, however, not fully optimized because as response to a suspicion message, the suspected node over-sends information. In the current implementation, the suspected node replies with its entire membership list which includes itself in a recent, alive state. Instead, a low-information ping and acknowledgement is sufficient for this purpose. Thus, Gossip with S could be achieved with no substantial overhead to Gossip without S given that the UDP protocol itself does not incur substantial overhead on each datagram.



As a tradeoff for the higher bandwidth we see when using suspicion, we now have a reduction in false positivity cases. With T_fail staying constant, our implementation of suspicion in which the suspected node immediately responds with its membership table allows it to respond without waiting for the next heartbeat. Since T_fail is not orders of magnitude larger than T_gossip, cases of false positivity occur naturally given a small enough network in which less dropped packets must occur for a false positive case to occur. That being said, as discussed in the earlier paragraph, Gossip without suspicion could be implemented with little overhead, in which case there is minimal tradeoff for this accuracy.



In terms of true failure detection, the implementation of Gossip with S and Gossip with S does not change the synchronous gossip heart-beating pattern. Instead, our implementation consists of the membership list recording an FSM for each node. The bandwidth increase is incurred by the membership list response by the suspected node in cases where it would otherwise not be gossiped to and require a response. Thus, there should be no significant different in the detection latency, which is represented by the bar charts even though the axes are a bit misleading. In practice, the latency still depends on the combination of the heartbeat frequency and T_failure.