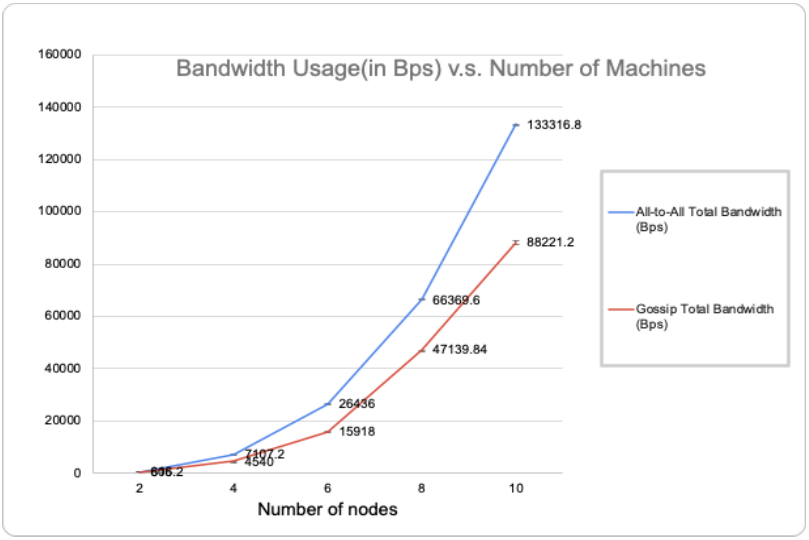
CS425-MP1-Report Authors: Dayue Bai(dayueb2), Yitan Ze(yitanze2)

Format of Messages

We defined our own message format based on Java String class. Each message sent among machines is serialized into a Java String and will be deserialized by the machine that receives the message. In this way, we can ensure that the message is platform-independent because Java is a platform-independent language. Self-defined messages start with a string indicating its action followed by specific details, e.g. “[Heartbeat]SerializedMembershipList” and “[JOIN]NodeID”.

Background Bandwidth Usage v.s. Number of Machines

Regardless of the failure detection protocol used, the background bandwidth increases monotonically as the number of machines increases, because more heartbeats are being sent among machines in one heartbeat period. The bandwidth usage of all-to-all is higher than that of gossip protocol, **especially** when there are more machines in the group due to the nature of these two different protocols. For instance, when there are only two nodes in the group, each node sends one heartbeat to the other node in each heartbeat period regardless of the protocol (same bandwidth). When there are 10 nodes in the group, each node sends heartbeats to 9 other nodes (all-to-all), while each node only sends heartbeats to 6 other nodes (gossip) with a sample ratio: 0.6.



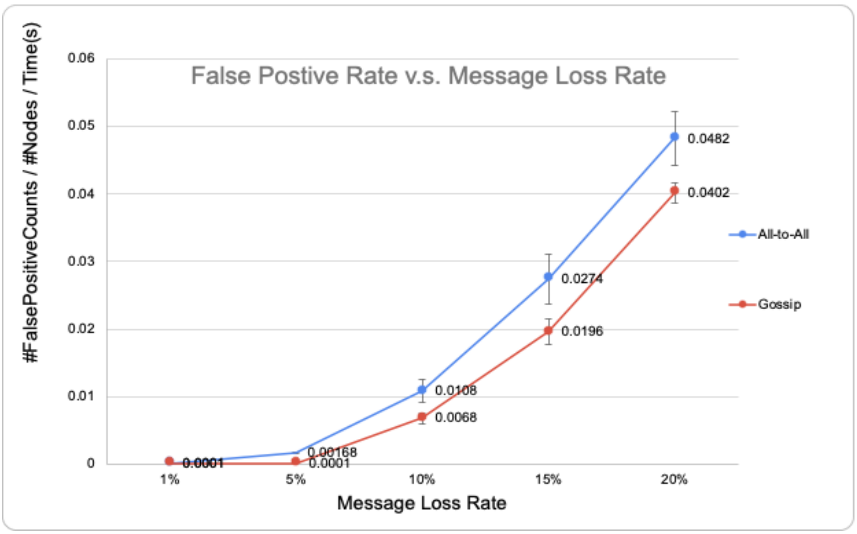
*E.g. when there are 10 machines, the **total** background bandwidth usage is 133316.8 Bps for All-to-All, and 88221.2 Bps for Gossip.*

Measurement:
For each data point in this plot, we measure the value for 5 times and take the average to plot the graph. Standard deviation bars (5 values’ standard deviation) are also plotted around each data point.

False Positive Rate v.s. Message Loss Rate

We calculated the false positive rate by calculating [false positive counts] / [number of nodes] / [time(in seconds)] and we simulate message loss rate by **randomly** dropping a certain precalculated amount of heartbeats. 10 VMs are used for experiments. For both all-to-all and gossip, heartbeat period = 500ms, T_timeout = 2000ms.

From the graph below, we can see that the false positive rate increases monotonically as the message loss rate increases. For all to all, if Node 1’s certain heartbeat packet is lost, some other nodes may mistakenly mark it failed when no heartbeat is received after T_timout. As the message loss rate increases, the probability of false positive cases also increases. The same reason also applies to Gossip. Given a certain message loss rate, the false positive count of all-to-all is always larger than that of the gossip protocol. In the Gossip-style protocol, each node merges its local membership list with the membership list received from other nodes through heartbeats. Thanks to its design, Gossip protocol can resist higher packet loss rate For example, there are 3 nodes: A, B, and C. Even if A’s heartbeat to C is lost, C can still update A’s timestamp if B sends new information about A to C. By contrast, in all-to-all protocol, C will mark A failed when it does not receive a heartbeat from A **directly** after T_timeout. Therefore, all-to-all protocol leads to a higher false positive rate than Gossip-style protocol, especially when message loss rate increases to 15% and 20%.



Comment: link to our original test data (includes 5 tests for each data point, average, standard deviation)