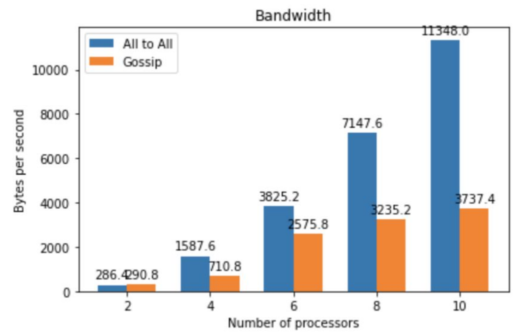
Seongwoo Kang (skang75) / HeonJang Lee (hl8)
MP1 Report

1. Background Bandwidth Usage

10 VMs, T_fail, T_timeout = 5 secs, K = 3

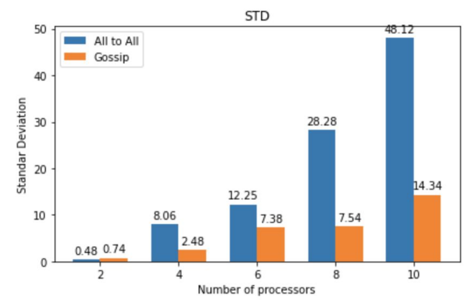


For All-to-All, we expected bandwidth to grow exponentially O(N^2) with the number of processors, and the data met our expectations.

For Gossip, we expected the bandwidth to grow exponentially O(KN) with the number of processors until <= # of processor = 4 (i.e. k+1). However, the bandwidth showed rapid growth until 6. Even though the number of receivers does not increase anymore, the size of the packet is relatively small, thus an increase in size of the packet affects the bandwidth crucially. After # of processors >= 6, it showed a linear growth (respect to packet size) as we expected.

STD increased with respect to its corresponding bandwidth as expected

```
ALL to ALL | 0 processors:  286.40 bps (std 0.48)
gossip     | 0 processors:  290.80 bps (std: 0.74)
ALL to ALL | 2 processors:  1587.60 bps (std 8.06)
gossip     | 2 processors:  710.80 bps (std: 2.48)
ALL to ALL | 4 processors:  3825.20 bps (std 12.25)
gossip     | 4 processors:  2575.80 bps (std: 7.38)
ALL to ALL | 6 processors:  7147.60 bps (std 28.28)
gossip     | 6 processors:  3235.20 bps (std: 7.54)
ALL to ALL | 8 processors:  11348.00 bps (std 48.12)
gossip     | 8 processors:  3737.40 bps (std: 14.34)
```



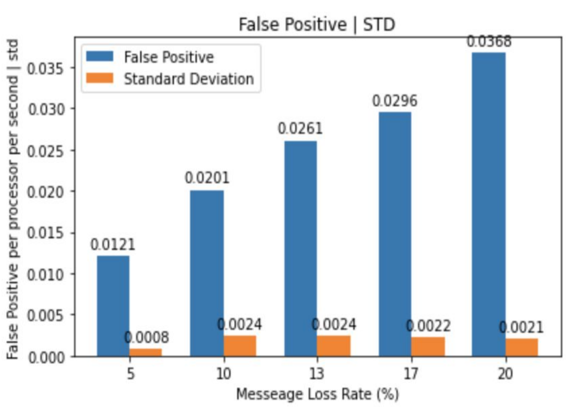
2. False Positive Rate

Number of Machines = 4, Timeout = 5, K = 1

For all-to-all, we have approximated the probability of FP to be extremely low. When we have attempted to detect FP using loss rate from 0% to 20%, we could not detect any FPs. This is due to the very low possibility of all machines failing to detect the same server simultaneously for five consecutive seconds to pass timeout.

$$FailProb_{ata} \approx (Loss)^{5(n-1)}$$

$$FailProb_{gossip} \approx \left(\frac{n-k}{n} + \frac{n}{k} * Loss\right)^{5(n-1)}$$



Using the formula we generated, our original parameters had extremely low probability of FP, so we used parameters that give higher FP probability. However, the result gave FPs higher than our expectation. We believe that this discrepancy is caused by our failure to account for the chained failures. When a processor is considered failed, a processor does not send packets to that processor. Thus, other processors are more likely to not receive enough information from that "Failed" processor, leading to more instances of FP.

```
Message Loss Rate:  05 % | False Positive/processor/second: 0.0121 bps (std 0.0008)
Message Loss Rate:  10 % | False Positive/processor/second: 0.0201 bps (std 0.0024)
Message Loss Rate:  13 % | False Positive/processor/second: 0.0261 bps (std 0.0024)
Message Loss Rate:  17 % | False Positive/processor/second: 0.0296 bps (std 0.0022)
Message Loss Rate:  20 % | False Positive/processor/second: 0.0368 bps (std 0.0021)
```