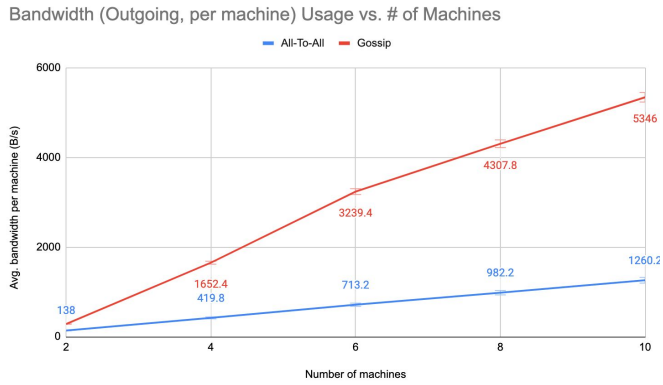
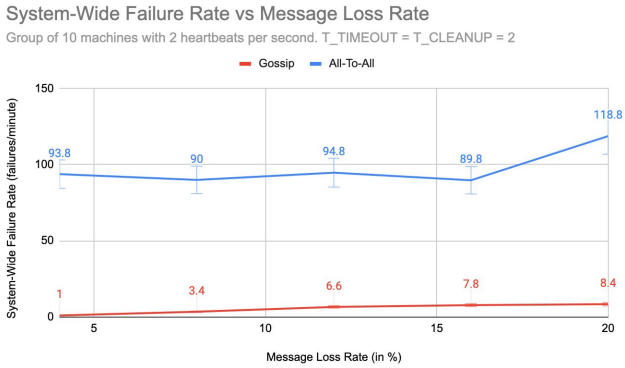
Ishaan Sehgal (netid: isehgal2) • Heesoo Yang (netid: heesooy2) • Asher Dale (netid: asdale2)

CS425 MP1 Report

We implemented our Distributed Group Membership Service using Golang, and for platform-independent messaging we used Google Protocol Buffers. When a machine runs the executable, it consists of a main thread that starts two “goroutines” (lightweight threads): one goroutine for listening for messages from other machines, and one goroutine for sending messages to other machines at a fixed interval. After starting the goroutines, the main thread listens for command-line messages from the user. A membership list is stored at each machine as a dictionary, with the machine ID (IP address concatenated with timestamp) as the keys and heartbeat information as the values. For gossip we send the entire membership list, and for all-to-all we just send heartbeats. In an attempt to modularize our code, we have split functionality into the following packages: networking, logging, membership, detector, and config.



This plot is what we expected because the gossip protocol has more background bandwidth usage (since we send the entire list), and the bandwidth usage for each machine increases linearly for both protocols. Although there are fewer messages sent in gossip mode, the size of the message scales linearly with the number of rows in the membership list. For all-to-all, the message size remains constant, but the number of messages sent increases linearly as the number of machines increases.



The difference in failure rates between the gossip protocol and the all-to-all protocol is what we expected because the all-to-all protocol is less accurate since it does not forward indirect heartbeat counters. As the message loss rate increased at a linear rate, it was expected and found that the gossip failure rate increased. For all-to-all, we attribute the lack of a steady slope to the inherent message loss rate in UDP on the VMs that we could not take into account. Data was collected by summing false detections across all machines at 1 minute intervals.