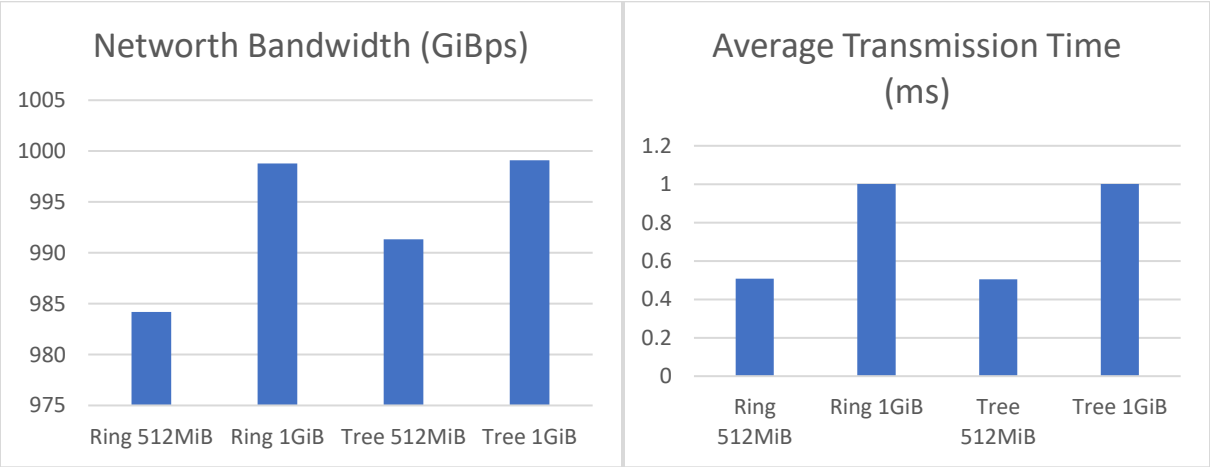
Assignment PA1

Michael Rivnak

COMP 3450

These two algorithms, ring and tree, function fundamentally differently. The ring algorithm is more of a peer-to-peer type where no single node has priority over any of the others, nor functions differently in how it sends and receives data. The tree algorithm, however, is a host-client type of structure, where the host is responsible for distributing data to all the other nodes, then collects the data back from each node as it completes its task. After some testing it was shown that each of these has the same network bandwidth, this makes sense as in this case it has no correlation to the algorithm. Although what is affected is the time it takes for each node to receive data as the tree algorithm requires significantly more steps. The following graphs show data transfer speeds with the two algorithms at a data size of 512MiB and 1GiB. It is interesting that the 512MiB sample both came to a lower speed, it is possible that this is due to the extra tasks associated with sending and receiving data is a larger percentage of the time taken by those processes than sending the data itself.



The communication bottleneck for these two applications is different but still significant. The bottleneck in this case is seen as the degree of difficulty that sending data between nodes presents. Of course, since no computations are being made in the case of the big data programs the data transmission will always be the bottleneck, so it is impossible to say that my program is free of bottlenecks. The token ring is considerably less incumbered by the bottleneck since it sends data linearly rather than having to report back to the “host” after every operation. Although since it functions in a ring it must travel to each node before the data returns to the original node. In the tree function the bottleneck is significant, data transfer between “client” nodes occurs at double the normal rate due to all the data being sent back to the host node first. Depending on how this is ultimately implemented it may be easier for specific nodes to receive data since it does not need to pass through every other node first, but in this implementation, this is not relevant.